

**Wałbrzyska Wyższa Szkoła
Zarządzania i Przedsiębiorczości
Studia Podyplomowe
INFORMATYKA DLA NAUCZYCIELI**

Dorota SKRABKA

**MODELOWANIE PROBLEMU I JEGO
ROZWIAZANIE PRZY WYKORZYSTANIU
TECHNIKI I TECHNOLOGII INFORMATYCZNEJ**

Praca napisana pod kierunkiem dr W. Pawlaka

**WAŁBRZYCH
2003**

Wstęp.....	4
1. Algorytmy wokół nas.	7
1.1. Algorytmy na przestrzeni wieków.....	7
1.2. Reprezentacje problemów i algorytmów.....	8
1.2.1. Zadanie rachunkowe jako przykład algorytmu.....	8
1.2.2. Słowny opis algorytmu.....	9
1.2.3. Opis algorytmu w postaci listy kroków.....	10
1.2.4. Schemat blokowy algorytmu.....	11
1.2.5. Drzewo algorytmu.....	12
1.2.6. Drzewo wyrażenia.....	12
1.2.7. Algorytm funkcji $f(x) = x / x $ utworzony w programie ELI.....	13
1.2.8. Algorytm funkcji $f(x)=x / x $ utworzony w programie Pascal.....	15
2. Algorytmy w korelacji z innymi przedmiotami.	16
2.1. Algorytmy w Podstawie programowej.....	16
2.2. Algorytmika w gimnazjum.....	17
2.2.1. Ścisłe sformułowanie sytuacji problemowych.....	19
2.2.2. Opisywanie algorytmów w języku potocznym.....	19
2.2.3. Zapisywanie algorytmów w postaci procedur.....	20
2.2.4. Algorytmy rekurencyjne.....	20
2.2.5. Rozwiązywanie zadań metodą zstępującą.....	21
2.2.6. Testowanie i ocena algorytmów. Własności algorytmów.....	21
2.3. Znaczenie algorytmów w programach nauczania.....	22
2.4. Ścieżki przedmiotowe.....	22
3. Problem reszty -przykład algorytmu zachłannego.	24
3.1. Algorytmy zachłanne.....	24
3.2. Problem reszty – jako przykładowy konspekt lekcji.....	24
3.2.1. Cele kształcenia.....	24
3.2.2. Treści problemowe.....	25
3.2.3. Rozwiązywanie problemów i podejmowanie decyzji.....	25
3.2.4. Działania.....	25
3.2.5. Stawiamy problem:.....	26
3.2.6. Dyskusja nad wyborem metody rozwiązania.....	32
3.2.7. Przykłady do zadań. Wnioski z nich wypływające.....	34

3.2.8.	Zachłanna metoda postępowania przy wydawaniu reszty.....	35
3.2.9.	Zachłanna strategia postępowania - algorytm postępowania.	35
3.2.10.	Algorytm wydawania reszty.....	36
3.2.11.	Zadania do wykonania przez uczniów w formie ćwiczeń.....	39
3.2.12.	Osiągnięcia.....	39
3.3.	Problem reszty – program zapisany w języku Pascal.....	40
3.3.1.	Opis zadania.....	40
3.3.2.	Zapis w programie Pascal – wydawanie reszty.....	41
Wnioski.	41
Literatura.	43

Wstęp

Od początków swojego istnienia człowiek zawsze upraszczał swoje czynności i działania, od: wzniesienia ognia, budowania piramid, wysyłania wiadomości, wychodzenia z labiryntu, optymalnego poruszania się po drogach, aż po sterowanie maszynami. Zwykle, pierwszym zamierzeniem w tym działaniu było osiągnięcie wyznaczonego celu w jakikolwiek sposób. Korzystał przy tym, często nieświadomie, z posiadanej już wiedzy i umiejętności zdobytych przy znajdowaniu rozwiązań innych problemów. Niemal jednocześnie z pojawieniem się pierwszych zadań do wykonania, człowiek zaczął sięgać po różne narzędzia, które miały mu w tym pomóc. Myśl plus pomoc (narzędzie) z czasem przeradzały się w technikę lub technologię. Również i my, gdy już potrafimy coś zrobić, zastanawiamy się, jak to można zrobić prościej, lepiej, mniejszym wysiłkiem, szybciej, a czasem - stać nas na luksus zrobienia czegoś pięknie, z większą elegancją i przyjemnością. Zastanawiamy się, jak już swoją, nabytą wiedzę przekazać innym. Zaczynamy budować przepis, krok po kroku, jak osiągnąć zamierzony cel lub jak rozwiązać problem. Zaczynamy budować algorytm¹. Możemy wybrać jedną z dróg: spróbować rozwiązać problem samodzielnie lub skorzystać z gotowego rozwiązania. W tym drugim przypadku, gdy wybiera się gotowe rozwiązanie, dobrze jest je zrozumieć. Najczęściej znajdujemy się jednak w sytuacji, gdy dla rozważanego przez nas problemu nie ma w pełni gotowego rozwiązania, a jeśli jest - to mamy kłopoty z jego zrozumieniem. Okazuje się, że najlepszą drogą do tego, by móc rozwiązywać problemy i rozumieć ich rozwiązania, jest poznanie sposobów prowadzących do ich otrzymania. Stąd podjęcia działań aby przygotować człowieka do poznania sposobów otrzymania rozwiązań, poznania ogólnych zasad i metod, które pozwolą zadanie zdefiniować, opisać ją dostępnymi parametrami, narzędziami określić sposób rozwiązania i wreszcie rozwiązać je. Zadania te realizuje współczesna szkoła. A zatem, aby rozwiązać problem uczeń musi, być może za pomocą nauczyciela, zrobić użytek ze swojej wiedzy, a często pogłębić ją w tym celu. Rozwiązanie danego problemu jest więc nie tylko wynikiem uczenia się, ale procesem rozwiązywania problemu, jest sposobem uczenia się. W konkretnym przypadku, interesuje nas znalezienie rozwiązania dla postawionego problemu (a ogólniej, stworzenie sytuacji problemowej). Za-

¹ Słowo algorytm pochodzi od brzmienia fragmentu nazwiska: Muhammad ibn Musa al-Chorezmi, arabskiego matematyka i astronoma, żyjącego na przełomie VIII i IX wieku. Jest on uznawany za prekursora metod obliczeniowych w matematyce. Od fragmentu tytułu jego dzieła pochodzi inne słowo, algebra. Upowszechnił on również stosowanie systemu dziesiętnego i posługiwanie się zerem.

sadniczy cel nauczania rozwiązywania problemów to: nabycie przez uczniów ogólnych zasad rozwiązywania problemów oraz ukształtowanie podstawy poszukiwania problemów i dążenia do ich rozwiązywania. Można to osiągnąć poprzez przygotowanie uczniów do posługiwania się współczesnymi „narzędziami” komputerem oraz technologią informacyjną². Nauczyciele mają i muszą stwarzać uczniom warunki do nabywania umiejętności poszukiwania, porządkowania i wykorzystywania informacji z różnych źródeł oraz posługiwania się w tym komputerem i technologią informacyjną³. To połączenie słów ma uzasadnienie we współczesnej postaci informacji i w sposobach korzystania z niej. Informacji towarzyszą nieustannie procesy i działania. Zarówno sam obiekt - informacja - w każdej chwili ulega zmianie (poszerzeniu, aktualizacji, dopisaniu powiązań, nowym interpretacjom itd.), jak i korzystanie z niej jest procesem. Nie tylko sięgamy po nią, jak po fragment zapisany w książce (np. w encyklopedii), ale - pisząc odpowiednią frazę i wydając polecenie dla komputerowego systemu wyszukiwania informacji - uruchamiamy proces jej uformowania w odpowiednim zakresie i postaci [1],[2].

Zatem odpowiedź formowana przez uczniów na postawione im pytanie, czy też rozwiązanie przez nich konkretnego problemu, przyjmuje w rzeczywistości postać chwilowego stanu tych procesów. Ten współczesny aspekt informacji pojawił się dzięki technicznym możliwościom - informacja i posługiwanie się nią staje się procesem dzięki stosowaniu szeroko rozumianej technologii informacyjnej. Należy to uwzględnić w procesie edukacji. Na czym więc ma polegać kształcenie się i nauczanie w środowisku o takich cechach, jak: nieograniczona zawartość i ciągłość zmian, w tym przypadku w odniesieniu do informacji? Odpowiedź jest prosta - w danej chwili powinniśmy umieć skorzystać z zatrzymanego na moment procesu związanego ze stanem interesującej nas informacji. A jaką przyjąć strategię poruszania się w nieograniczonym uniwersum informacji, podlegającym ciągłym zmianom? Sugestia jest również prosta: reakcją na poznanie procesu związanego ze stanem informacji powinno być również uruchomienie w sobie procesu związanego z pozyskiwaniem form,

² Coraz popularniejsze staje się w państwach wspólnoty Europejskiej określenie technologia informacyjna i komunikacyjna, w którym oba aspekty, informacja – jako obiekt podlegający działaniu, i komunikacja – jako przeznaczenie informacji i cel działania technologii, zostają zrównane.

³ Technologia informacyjna (TI) jest to zespół środków (czyli urządzeń, takich jak komputery i ich urządzenia zewnętrzne oraz sieci komputerowe) i narzędzi (czyli oprogramowanie), jak również inne technologie (takie, jak telekomunikacja), które służą wszechstronnemu posługiwaniu się informacją. TI obejmuje więc swoim zakresem m.in.: informację, komputery, informatykę i komunikację. Współczesna technologia informacyjna wyrosła na bazie zastosowań komputerów, a jej decydujące znaczenie dla życia społeczeństw upoważnia do zdefiniowania końca XX wieku jako ery informacji i jej technologii.

w których występuje informacja, oraz mechanizmów i metod korzystania z niej. **Stąd podstawowy cel mojej pracy.**

Uczeń powinien poznać następujące etapy rozwiązywania problemów:

1. sformułowanie zadania,
2. określenie danych wejściowych,
3. określenie celu - wyniku,
4. poszukiwanie metody rozwiązania - algorytmu,
5. przedstawienie algorytmu w postaci:
 - a. opisu słownego,
 - b. listy kroków,
 - c. schematu blokowego,
 - d. jednego z języków programowania,
6. analiza poprawności rozwiązania,
7. testowanie rozwiązania dla różnych danych - ocena efektywności przyjętej metody.

Nadrzędnym celem mojej pracy jest pomoc w wypracowaniu przez uczniów sposobów efektywnego poruszania się wśród zbiorów informacji. Obecnie, informacje i posługiwanie się nimi są nieodłącznym elementem każdej dziedziny. Stawiam tezę, do której staram się przekonać pozostałych nauczycieli, że: integralną cechą informacji jest jej uporządkowanie, w przeciwnym razie ... nie jest to informacja. Jeśli tak jest, to podstawowym przygotowaniem ucznia do życia w erze informacji jest nabycie umiejętności takiego postępowania z informacją (uporządkowaną oczywiście), by korzystał (w operacjach) z tej jej cechy, czyli z uporządkowania, nie psuł jej i ewentualnie naprawiał, gdy ulega zniszczeniu.

W swojej pracy skupię się przede wszystkim na algorytmice⁴ występującej w III etapie kształcenia (gimnazjum), której zakres wynika z zapisów umieszczonych w podstawie programowej przedmiotu informatyka⁵. Chcę pokazać, że zajęcia o algorytmach można tak poprowadzić, by były interesujące i przystępne dla wszystkich uczniów, by wyzwalaly „moce twórcze” nie tylko wśród tych, którzy interesują się konkursami informatycznymi lub matematycznymi. Chcę pokazać, że uporządkowane postępowanie z posiadaną informacją (algorytmem) zawsze przyniesie sukces.

⁴ Algorytmika jest nazwą dziedziny zajmującej się algorytmami i ich własnościami. Po raz pierwszy tego terminu użył Dawid Harel w tytule swojej książki *Rzecz o istocie informatyki – algorytmika*. Powstała ona na kanwie pogadank o algorytmach, prowadzonych przez autora w izraelskim radiu.

⁵ MEN, 15 luty 1999 r. Dalej ten dokument jest w skrócie nazywany Podstawą programowe.

1. Algorytmy wokół nas.

1.1. Algorytmy na przestrzeni wieków.

Budowanie algorytmów jest tak stare, jak dążenie człowieka do rozwiązywania problemów lub zdobywania wyznaczonych celów. Człowiek, jak sięgnąć pamięcią, zawsze starał się ułatwić sobie życie oraz ulepszać metody realizacji swoich zamierzeń. Najstarszy algorytm to algorytm Euklidesa, który ma ponad 2000 lat. Wiele innych algorytmów i metod ich tworzenia opracowano przed pojawieniem się pierwszych komputerów. W dużej części były one dziełem matematyków i inżynierów, którzy opracowywali metody rozwiązywania problemów rachunkowych. Niektóre z tych metod są opisane dalej. Sporo współczesnych algorytmów korzysta ze zdobyczy przeszłości. Co więcej, okazało się, że często pomysły naszych przodków charakteryzowała olbrzymia intuicja i podane przez nich algorytmy są z powodzeniem stosowane również w dobie komputerów.⁶

Równocześnie z tworzeniem algorytmów człowiek starał się budować urządzenia, które pomogłyby mu wykonywać obliczenia. Pierwsze takie urządzenia były liczydłami - konstruowali je Chińczycy, Rzymianie, Arabowie. W XVII wieku zaczęto budować urządzenia, które nazwalibyśmy dzisiaj kalkulatorami - ich twórcami byli m.in. Blaise Pascal (1623-1662) i Gottfried W. Leibniz (1646-1716). Od początku XIX wieku rozpoczyna się era maszyn budowanych na zasadach, które odnajdujemy w obecnych komputerach⁷ - pierwszym wielkim konstruktorem takich maszyn był Charles Babbage (1791-1871). Jednak twórcą komputera w dzisiejszym sensie jest Konrad Zuse (1910-1995) - pierwszą swoją maszynę zbudował on jeszcze przed II wojną światową [3],[6].

⁶ Za prekursora algorytmów komputerowych jest uznawana powszechnie Ada Augusta (1815-1852), hrabina Lovelace, córka Byrona. Uważa się ją również za pierwszą programistkę komputerów - to jej imieniem nazwano jeden z nowoczesnych języków programowania wysokiego poziomu, język Ada. Zachwycona konstrukcją analitycznej maszyny Ch. Babbage'a uważała, że będzie ona „tkać wzory algebraiczne, jak krosna Jacquarda tkają liście i kwiaty”. Przełomowe znaczenie tej maszyny upatrywała „w możliwości wielokrotnego wykonywania przez nią danego ciągu instrukcji, z liczbą powtórzeń z góry zadaną lub zależną od wyników obliczeń”. Dzisiaj tak określa się podstawowe cechy algorytmów komputerowych.

⁷ Jeszcze zanim zbudowano pierwszy komputer naukowcy snuli rozważania o maszynie myślącej. Powstała wtedy koncepcja automatu wykonującego instrukcje - sławna po dziś dzień maszyna Turinga. Wkrótce John von Neumann przedstawił schemat architektury komputera szeregowego. W laboratoriach amerykańskich zbudowano pierwszy komputer ENIAC, który działał na lampach elektronowych i przełącznikach. Z czasem możliwości komputerów sprawiły, że próby stworzenia sztucznej inteligencji zaczęły przynosić efekty.

1.2. Reprezentacje problemów i algorytmów.

Reprezentacje algorytmów są nie tylko ich opisami, ale często wykorzystuje się je przy konstruowaniu algorytmów oraz badaniu własności wykonywanych przez nie obliczeń. Żaden ze sposobów zapisywania algorytmów nie jest faworyzowany. Ze względu na objętość niniejszej pracy, posłużę się tylko wybranymi algorytmami. Dany algorytm jest opisany w najbardziej odpowiedni dla niego sposób, który zależy od: rodzaju wykonywanych w nim operacji oraz możliwości wybranej reprezentacji. Na przykład, nie każdy algorytm można zapisać w postaci drzewa obliczeń lub drzewa wyrażeń⁸. Podobnie, ze względu na ograniczoność miejsca na ekranie monitora i brak złożonych struktur danych, nie dla wszystkich algorytmów można utworzyć schemat w odpowiednich programach. Jeśli z kolei planuje się wykonywanie eksperymentów komputerowych z algorytmem, to właściwym opisem może być schemat w programie np. ELI, program komputerowy a, dla prostszych algorytmów, schemat blokowy. Poszczególne rodzaje opisów algorytmów wprowadzę tylko w takim zakresie, w jakim będą mi potrzebne w konkretnym przypadku. Ponieważ celem pracy jest przedstawienie algorytmów w procesie ich konstruowania, mniejszą wagę przywiążę do ich opisów w postaci programów komputerowych czy opisów samych algorytmów. Chcę w ten sposób osiągnąć jeszcze jeden cel - by o algorytmach myśleć jak o sposobach rozwiązywania problemów, niezależnych od wybranej reprezentacji oraz systemu programowania, w którym algorytm ma być wykonany za pomocą komputera. Algorytmiczne myślenie można bowiem kształtować niezależnie od programowania komputerów, chociaż każdy program komputerowy jest zapisem jakiegoś algorytmu.

1.2.1. Zadanie rachunkowe jako przykład algorytmu.

Posłużymy się następującym przykładem zadania rachunkowego, dla którego utworzymy algorytm i przedstawimy różne jego reprezentacje.

Oblicz wartość funkcji:

$$f(x) = \frac{x}{|x|} \quad (1.1)$$

⁸ Drzewo w opracowaniach informatycznych jest rysowane na ogół do góry nogami, z korzeniem u góry. Stosowane nazewnictwo elementów drzewa jest wzięte z dendrologii – korzeń, wierzchołki końcowe nazywają się liśćmi, a krawędzie, czyli połączenia w drzewie - gałęziami.

dla dowolnego rzeczywistego x , gdzie $|x|$ oznacza funkcję - wartość bezwzględna, zwaną również modułem. Naszym zadaniem jest podanie algorytmu, czyli sposobu obliczania wartości tej funkcji. Już na tym prostym przykładzie widać, że przed przystąpieniem do opisanego sposobu rozwiązania zadania, musimy najpierw zadbać o dokładne jego sformułowanie. W informatyce używa się pojęcia specyfikacja na oznaczenie dokładnego opisu zadania, które ma być wykonane lub problemu, który ma być rozwiązany. Na ogół, specyfikacja ma postać wyszczególnienia, w którym są wymienione dane dla zadania i warunki jakie muszą one spełniać, oraz wyniki określone poprzez dane wejściowe, a ściślej - jaki jest związek wyników z danymi. Aby podać specyfikację zadania polegającego na obliczaniu wartości funkcji (1.1), musimy uzupełnić jej opis o dziedzinę, czyli dla jakich wartości x funkcja ta ma określone wartości. Z postaci wzoru wynika, że x nie może być równe zero. Ustalmy jednak dodatkowo, że wartością funkcji f w zerze jest 0. Zatem nasza funkcja jest określona dla wszystkich liczb rzeczywistych x . Możemy już teraz podać specyfikację rozwiązywanego zadania:

Obliczanie wartości przykładowej funkcji

Dane: Dowolna liczba rzeczywista x .

Wynik: Wartość funkcji $f(x)$ podanej wzorem (1.1), jeśli x jest różne od zera i 0 - w przeciwnym przypadku.

W tej specyfikacji wyszczególniliśmy, że daną w problemie jest jedna liczba rzeczywista, a wartością - również liczba rzeczywista. Dodatkowo, z określenia wyniku wiemy, że jeśli dana jest różna od zera, to wynik jest określony wzorem (1.1), a jeśli $x = 0$, to wynikiem jest również 0.

W następnych punktach tego rozdziału wprowadzę różne reprezentacje algorytmu rozwiązywania zadania o tej specyfikacji.

1.2.2. Słowny opis algorytmu.

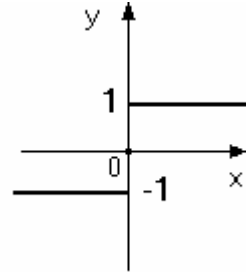
Słowny opis algorytmu jest na ogół jego pierwszym, mało ścisłym opisem. Rozpoczyna się często dyskusją, w jaki sposób można rozwiązać postawione zadanie, i służy wyrobieniu pewnej intuicji oraz ukierunkowaniu rozważań na właściwe sposoby i techniki przydatne w rozwiązaniu. Dyskusję o sposobie obliczania wartości funkcji danej wzorem (1.1) rozpoczęliśmy już w poprzednim punkcie. Wykorzystując znana definicję funkcji - moduł wartość bezwzględna $|x|$, specyfikacja naszego przykładu przyjmie następującą postać:

Obliczanie wartości przykładowej funkcji

Dane: Dowlolna liczba rzeczywista x

Wynik: Wartość funkcji $f(x)$ określonej następującym wzorem:

$$f(x) = \begin{cases} -1, & \text{dla } x < 0 \\ 0, & \text{dla } x = 0 \\ 1, & \text{dla } x > 0 \end{cases} \quad (1.2)$$



Rys. 1.1 Wykres funkcji (1.3)

1.2.3. Opis algorytmu w postaci listy kroków.

Lista kroków jest jednym z najczęściej stosowanych, dokładnych sposobów opisywania obliczeń oraz ich kolejności. Poszczególne kroki zawierają opis operacji, które mają być wykonane przez algorytm. Mogą w nich również wystąpić polecenia związane ze zmianą kolejności wykonywania kroków lub polecenia zakończenia algorytmu. Przyjmuje się w tym opisie, że kolejność wykonywania poszczególnych kroków jest zgodna z kolejnością ich wypisania, z wyjątkiem sytuacji, gdy jednym z poleceń w kroku jest przejście do wykonania kroku o podanym numerze. Dla ścisłości opisu, między tytułem algorytmu a listą jego kroków podajemy specyfikację problemu rozwiązywanego przez ten algorytm. Dodatkowo, w nawiasach $\{ i \}$ umieszczamy uwagi nie będące częścią algorytmu, a jedynie komentujące jego przebieg i pomagające zrozumieć wykonywane polecenia i ich efekty. Opis algorytmu w postaci listy kroków określa wykonywane w algorytmie działania i ich kolejność.

Dla przykładowego problemu, algorytm w postaci listy kroków zapisujemy następująco:

Algorytm obliczania wartości przykładowej funkcji (1.2)

Dane: Dowlolna liczba rzeczywista x .

Wynik: Wartość funkcji $f(x)$ określonej wzorem (1.2).

Krok 0. Wczytaj wartość danej x .

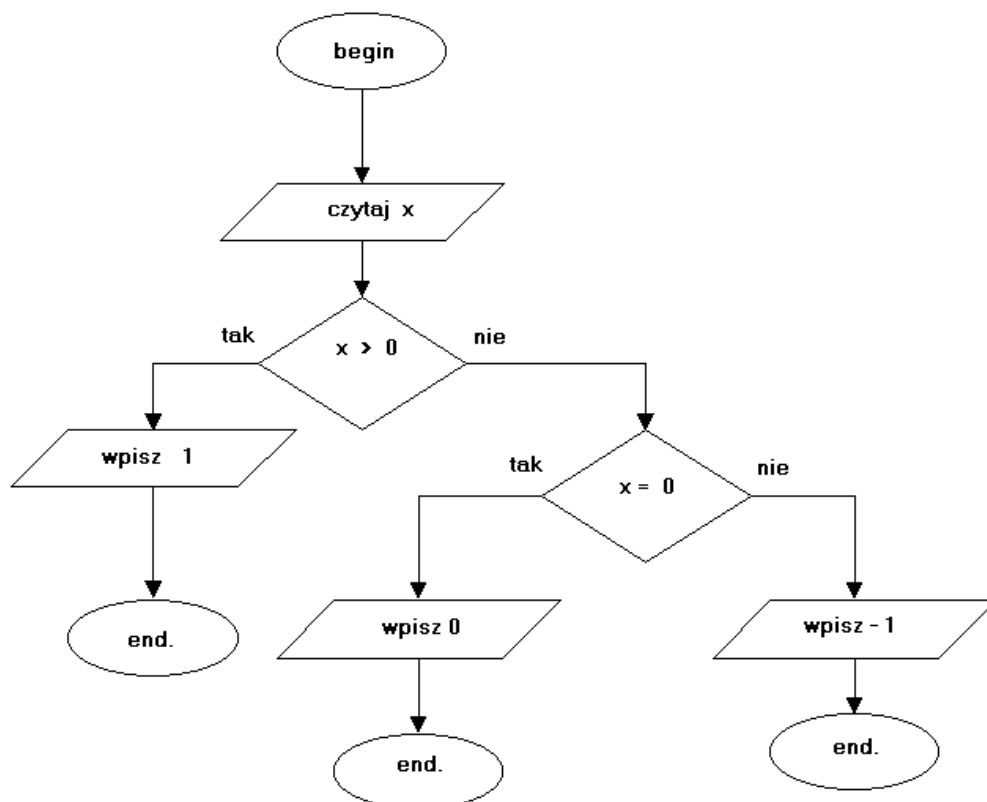
Krok 1. Jeśli $x > 0$, to $f(x) = 1$. Zakończ algorytm.

Krok 2. {W tym przypadku $x \leq 0$ } Jeśli $x = 0$, to $f(x) = 0$. Zakończ algorytm.

Krok 3. {W tym przypadku $x < 0$.} Mamy $f(x) = -1$. Zakończ algorytm.

1.2.4. Schemat blokowy algorytmu.

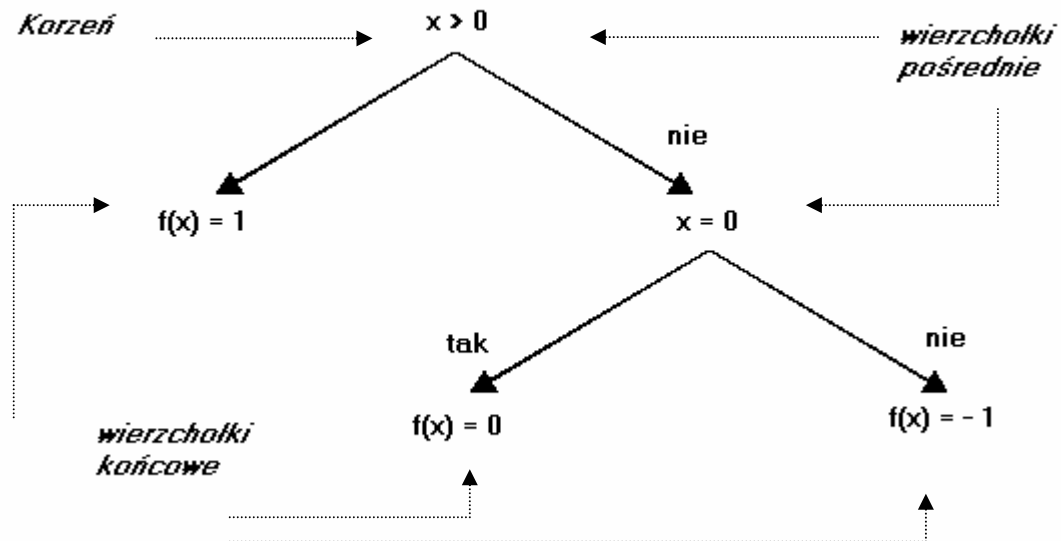
Schemat blokowy jest jednym z najbardziej popularnych, graficznych sposobów przedstawiania algorytmów. Składa się z klatek oraz połączeń między nimi. W klatkach są zapisywane operacje, które mają być wykonane, a połączenia wyznaczają kolejność ich wykonywania. Często stosuje się różne kształty klatek dla odróżnienia rodzajów operacji w nich zawartych. Schematy blokowe można budować odręcznie, można również konstruować je posługując się oprogramowaniem, które to umożliwia. Poniżej przedstawiono schemat blokowy algorytmu obliczania wartości funkcji (1.2)



Rys. 1.2 Schemat blokowy algorytmu obliczania wartości funkcji (1.2)

1.2.5. Drzewo algorytmu.

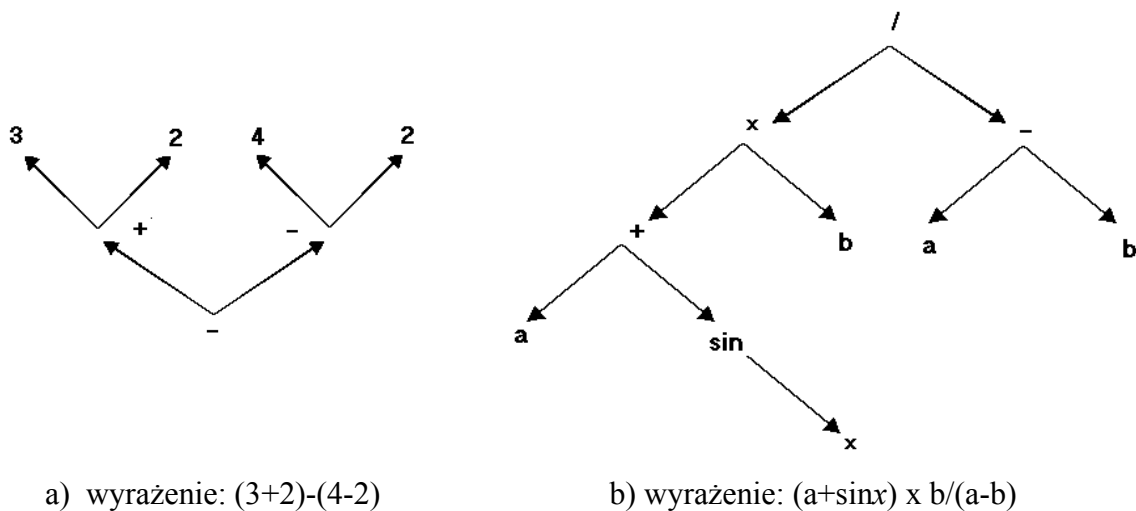
Drzewo algorytmu, nazywane również drzewem obliczeń, jest szczególnym rodzajem schematu blokowego, który przyjmuje postać drzewa. Oznacza to, że każde dwie drogi obliczeń w takim schemacie mogą mieć tylko początkowe fragmenty wspólne, ale po rozejściu już się nie spotykają. Schemat blokowy na rys. 1.2 ma właśnie postać drzewa algorytmu. Najczęściej drzewa algorytmów przedstawia się, upraszczając nieco ich postać graficzną - na rys. 1.3 jest pokazane drzewo odpowiadające schematowi z rys. 1.2. Ten sposób przedstawiania algorytmów jest bardzo dogodny do ich analizy, a zwłaszcza do wyznaczania liczby wykonywanych operacji.



Rys. 1.3 Drzewo algorytmu obliczania wartości funkcji (1.3), odpowiadające schematowi blokowemu z rys. 1.2.

1.2.6. Drzewo wyrażenia.

Drzewo wyrażenia ma ograniczone zastosowania jako reprezentacja algorytmu - jest wykorzystywane przede wszystkim do obliczania wartości wyrażeń. Z tą reprezentacją wyrażeń spotykamy się w szkole bardzo wcześnie, np. podczas nauki obliczania wartości wyrażenia arytmetycznego zawierającego nawiasy.



Rys. 1.4 Przykładowe drzewa wyrażeń.

Na rysunku 1.5a jest przedstawiony przykład drzewa rysowanego tradycyjnie, z korzeniem u dołu. W drzewie można zapisać dowolne wyrażenie. W takich drzewach argumenty wyrażeń znajdują się w liściach, a działania- w wierzchołkach pośrednich. Obliczanie wartości wyrażenia zapisanego w drzewie niejako spływa od liści, gałęziami ku korzeniowi (w dół lub do góry, w zależności od sposobu rysowania drzewa).

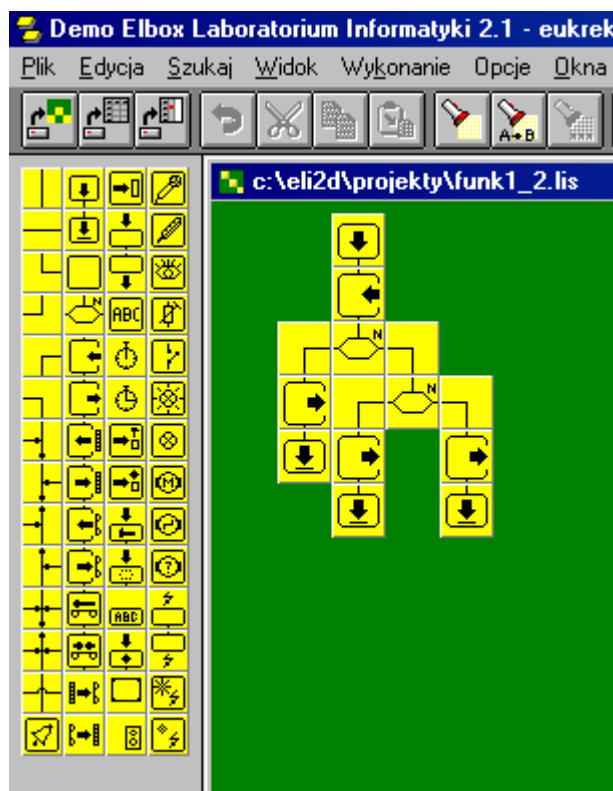
1.2.7. Algorytm funkcji $f(x) = x / |x|$ utworzony w programie ELI.

Program ELI będący jednym z wielu programów edukacyjnych, służy do konstruowania i analizowania algorytmów. W odróżnieniu od klasycznych schematów blokowych, w programie tym można budować algorytmy zawierające podprogramy, w tym również podprogramy rekurencyjne⁹ [4].

W programie ELI, algorytmy są konstruowane z gotowych klocków, które odpowiadają podstawowym krokom wykonywanym w algorytmach, takim jak: początek i zakończenie algorytmu, wprowadzanie danych i wyprowadzanie wyników, obliczanie wartości wyrażenia, instrukcja warunkowa (czyli rozgałęzienie w algorytmie), czytanie danych z tablicy lub z taśmy (czyli z pliku), zapisywanie danych w tablicy lub na taśmie, wywoływanie procedury (czyli podprogramu). Klocki służą jedynie do oznaczenia wybranych fragmentów algorytmów. Właściwe funkcje wykonywane przez nie w algorytmie należy wpisać w oknach dialogowych. Takie okno pojawia się na ekranie po naprowadzeniu

⁹ Rekurencja- sposób postępowania, a dokładniej – sposób zapisywania pewnych działań.

wskaźnika myszy na wybrany klocek i naciśnięciu prawego przycisku myszy. I tak, w klocek początku algorytmu można wpisać informację o algorytmie, w klocek wprowadzenia danej - podać nazwę zmiennej, której ma być przypisana wczytywana wartość, w klocek warunkowym warunek, od którego zależy dalsze działanie, w klocek wyprowadzenia wyniku - wyrażenie, którego wartość jest wyświetlana na ekranie, a w klocek zakończenia algorytmu – informację (komentarz), która będzie się ukazywać, gdy algorytm kończy działanie. Komentarz - zawiera ono informacje związane z kolejno wykonywanymi klockami. Śledząc informacje wyświetlane w tych oknach, można się zapoznać z pełnym przebiegiem algorytmu. Ich zawartość może również stanowić fragment dokumentacji tworzonej z przebiegu działania algorytmu, gdyż można ją przechować w pliku, a następnie: odpowiednio zredagować, włączyć do innych dokumentów i wydrukować.



Rys. 1.5 Zrzut ekranu utworzonego w programie ELI. Realizacja funkcji $f(x) = x / |x|$.

1.2.8. Algorytm funkcji $f(x)=x/|x|$ utworzony w programie Pascal.

Program to dla komputera zestaw instrukcji co zrobić w danym wypadku. Jednak musi być on zrozumiały dla komputera czyli napisany w jego języku. Jako, że język ten to same jedyńki i zera trudno było by się człowiekowi dogadać z maszyną. Potrzebny jest tłumacz czyli kompilator, który znając dany język może przełożyć go na język komputera. Programista uczy się więc rozmawiać z kompilatorem poznając jego język. Kompilatorów Pascala jest co najmniej kilka. Chyba najbardziej znanym jest Pascal czy Turbo Pascal. Ponieważ tematem pracy nie jest nauka programowania w języku Pascal, stąd podam już „gotowy” program napisany w języku Pascal do obliczania wartości funkcji (1.2).

```
program obliczanie wartosci funkcji  $f(x)=x/|x|$  ;  
    {program czyta liczbe x i drukuje wartosc funkcji  $f(x)=x/|x|$ .}  
    var x:real ;  
begin  
    read (x) ;  
    if x>0 then write (1)  
    else  
    if x=0 then write (0)  
    else write (-1)  
end. {wartosci funkcji  $f(x)=x/|x|$ }
```

Zainteresowanych do nauki programowania w języku Pascal odsyłam np. na stronę internetową: www.programming.enter.net.pl/pascal.html czy: www.winic.prv.pl

2. Algorytmy w korelacji z innymi przedmiotami.

2.1. Algorytmy w Podstawie programowej.

Algorytmy, w sposób jawny, najczęściej występują w programach szkolnych dotyczących edukacji informatycznej. Doszukać się ich można również, zwłaszcza po lekturze programów nauczania i podręczników, w zapisach dotyczących edukacji matematycznej, fizycznej oraz technicznej. Na podstawie dwóch programów nauczania, podręczników, zeszytów ćwiczeń i poradników metodycznych, można się zorientować, że po raz pierwszy, uczniowie słyszą w szkole o algorytmach na lekcjach matematyki. Czy jest to właściwe użycie pojęcia algorytm oraz na ile na lekcjach matematyki uczniowie są rzeczywiście wprowadzani do myślenia algorytmicznego oraz rozwiązywania problemów w postaci algorytmicznej, będzie przedmiotem moich rozważań. Oczywiście zaakcentuje tylko te najważniejsze, w mojej ocenie, spostrzeżenia. Skupię uwagę przede wszystkim na algorytmice występującej na III etapie kształcenia, czyli w gimnazjum. W zakresie programowej dla przedmiotu informatyka znaleźć można następujące sformułowania [5]:

Treści:
<ul style="list-style-type: none">▪ Rozwiązywanie problemów w postaci algorytmicznej. Algorytmy wokół nas, przykłady algorytmów rozwiązywania problemów praktycznych i szkolnych. Ścisłe sformułowanie sytuacji problemowych. Opisywanie algorytmów w języku potocznym. Zapisywanie algorytmów w postaci procedur, które może wykonać komputer. Przykłady algorytmów rekurencyjnych. Rozwiązywanie umiarkowanie złożonych zadań metoda zstępującą. Przykłady testowania i oceny algorytmów.
Osiągnięcia:
<ul style="list-style-type: none">▪ Rozwiązywanie umiarkowanie złożonych problemów przez stosowanie poznanych metod algorytmicznych.

Edukacja, jak każdy obszar działalności człowieka w społeczeństwie, we wszystkich swoich sferach powinna uwzględniać warunki, w których się odbywa, w tym również techniczne warunki życia społeczeństwa. Współczesna technologia informacyjna, która wyrosła na komputerach, ma obecnie olbrzymi wpływ na życie człowie-

ka i całych społeczeństw. Powinno to znaleźć odbicie w procesie wychowania i nauczania, by właściwie przygotować przyszłe pokolenia świadomych odbiorców i użytkowników tej technologii, przyszłych obywateli społeczeństwa informacyjnego.

Technologia informacyjna (TI) służy wszechstronnemu posługiwaniu się informacją i komunikowaniu się. Informacja w nauczaniu może pochodzić z dowolnej dziedziny kształcenia, być wykorzystywana w celach poznawczych i wychowawczych, a dzięki możliwościom współczesnej technologii informacyjnej - znajdować się w dowolnym miejscu na Ziemi. Z technologią informacyjną jest związana jedna z głównych umiejętności kształconych u uczniów w szkole: poszukiwania, porządkowania i wykorzystywania informacji z różnych źródeł oraz efektywnego posługiwania się technologią informacyjną.

Technologia informacyjna może wspomagać i wzbogacać wszechstronny rozwój uczniów, poprzez udział w realizacji większości zadań stawianych przed szkołą przez Podstawę programową, i zwiększać możliwości rozwijania umiejętności uczniów.

W zakresie ogólnych zadań szkoły, technologia ta wspomaga zdobywanie wykształcenia oraz głębsze poznawanie i rozumienie świata przez umożliwienie i ułatwienie docierania do rzeczywistych zasobów informacji. Wzmaga tym samym twórczą aktywność uczniów, pomaga w ujawnianiu i rozwijaniu zainteresowań oraz kierowaniu własnym rozwojem. Wzbogaca również sposoby i umiejętności porozumiewania się. Jej nieograniczone możliwości komunikacyjne mogą być wykorzystane do wzmocnienia u uczniów poczucia przynależności do różnych grup społecznych, jak również sprzyjać samoidentyfikacji narodowej i kulturowej. Kształtowanie właściwego korzystania z tej technologii to również wyrabianie świadomości bezpiecznego posługiwania się techniką. Niesie w sobie także aspekt wychowawczy, poprzez przygotowanie do respektowania norm etycznych i prawnych, poszanowania własności osobistej i intelektualnej oraz wyrabianie postawy otwartości na świat i tolerancji.

2.2. Algorytmika w gimnazjum.

Od wyboru sytuacji problemowej i algorytmu zależy powodzenie całych zajęć. Odpowiednim problemem możemy zainteresować i umotywować uczniów do działania, jaki i zrealizować założony przez nas cel – wprowadzenie wybranego sposobu rozwiązywania, a dokładniej - techniki algorytmicznej. Oto przykładowe sytuacje problemowe

z otoczenia uczniów wraz z podpowiedziami, które mogą służyć do wyprowadzenia algorytmów i algorytmicznych technik rozwiązywania problemów, nie tylko na zajęciach informatyki:

- a) wydawanie reszty w sklepie (i wiążące się z tym częstotści występowania monet na rynku) w postaci najmniejszej liczby monet, działanie automatów na monety; część lekcji może mieć charakter praktyczny i odbyć się na przykład w ... sklepie [3];
- b) organizowanie rozgrywek szkolnych – zaplanowanie rozgrywek szkolnych, które trwają najkrócej i mają wyłonić trzy najlepsze drużyny;
- c) dobór uczniów w pary według osobistych preferencji; mogą to być klasyczne pary taneczne albo dobór uczniów, by po dwóch pracowali przy komputerze, celem jest dobór się w pary, posługując się ... np. siecią komputerową);
- d) wybór trasy wycieczki – np. zaplanowanie wycieczki, która przebiegałaby po zadanych miejscach najkrótszą trasą;
- e) zachłanne pakowanie plecaka - wykorzystanie racjonalności żywienia w sytuacji wybierania się na wycieczkę z plecakiem;
- f) poszukiwanie z nawrotami – np. zajęcia dla uczniów, polegające na ustawianiu na szachownicy największej liczby nieatakujących się królowych; możliwość wykorzystania przy tym rzeczywistych plansz i programów komputerowy; poznanie metody prób - i - błędów, celem tu będzie wyrobienie pewnej systematyczności w poszukiwaniach;
- g) jednoczesne znajdowanie najmniejszej i największej liczby – np. uczniowie dzielą się sprawiedliwie wyciętymi drzewami, a polega to na tym, że kolejna osoba bierze spośród pozostałych drzew dwa, jedno o najmniejszej a drugie o największej kubaturze;
- h) rozwiązywanie układu dwóch równań liniowych - np. wyznaczenie momentu spotkania się dwóch rowerzystów, wyruszających o różnych godzinach i jadących z różną prędkością;

Każdy zaproponowany problem do rozwiązania można realizować na oddzielnych przedmiotach, zajęciach. Nie możemy kształtować wśród uczniów przeświadczenia, że algorytmy to tylko informatyka, matematyka. Treści problemowe powinniśmy zawsze realizować poprzez:

- a) ściśle sformułowanie sytuacji problemowych,
- b) opisanie problemu w postaci algorytmu,
- c) zapisanie algorytmów w postaci procedur, które może wykonać komputer,

2.2.1. Ścisłe sformułowanie sytuacji problemowych.

Każda z przytoczonych sytuacji problemowych wymaga najpierw precyzyjnego sformułowania, uwzględnienia wszystkich warunków, przedyskutowania znaczenia różnych czynników, również tych pobocznych, które mogą mieć wpływ na ostateczną postać rozwiązywanego problemu, a później - na jego rozwiązanie.

W wielu przypadkach, te sformułowania mogą być podane w języku sytuacji problemowej, i nie należy z tego rezygnować na korzyść np. liczb lub zapisu matematycznego. Najprościej formułuje się problemy matematyczne, które polegają na zastosowaniu wzoru. Jednak wymienione wyżej problemy nie są „czysto” matematyczne, chociaż algorytmy ich rozwiązywania, sprowadzone do postaci komputerowej, są wykonywane (w komputerze) na liczbowej reprezentacji danych.

Specyfika sytuacji problemowej i jej język powinny być utrzymywane na zajęciach jak najdłużej. Łatwiej wtedy uzasadnić formułowane założenia i wyciągane wnioski, a co najważniejsze - ostateczne rozwiązanie rozważanego problemu powinno odnosić się do rzeczywistej sytuacji problemowej, od wprowadzenia której rozpoczęły się zajęcia. Powiązanie sytuacji „wokół ucznia” z jej rozwiązaniem - ma decydujące znaczenie dla efektów nauczania [7].

2.2.2. Opisywanie algorytmów w języku potocznym.

Jest to na ogół pierwszy opis metody rozwiązywania i powinien być sformułowany w języku sytuacji problemowej. W kolejnym etapie rozwiązywania może on przyjąć postać listy kroków. Słowne opisy algorytmów mogą posłużyć do wydzielania fragmentów, które odpowiadają podstawowym konstrukcjom algorytmicznym, takim jak: iteracja (powtarzanie), rozgałęzienie (warunek), procedura (podprogram), oraz technikom rozwiązywania, np.: zachłanność, programowanie dynamiczne. W ten sposób, na przykładach rozwiązań rzeczywistych sytuacji, uczeń może poznać znaczenie poleceń: „powtórz ileś razy”, „powtarzaj, aż spełniony będzie określony warunek”, „powtarzaj, dopóki jest spełniony dany warunek”. Jeśli jest spełniony warunek, „to zrób to, a jeśli nie, to zrób tamto”, „wykonaj ten krok dla tych danych” itd. Uczeń poznaje także sposoby i efekty postępowania, nakazujące: „wykonaj najlepszy możliwy ruch”, „uwzględnij wszystkie możliwe sytuacje” itd. Te sformułowania mogą mieć różny stopień precyzji,

w zależności od przyjętego sposobu zapisywania rozwiązań w postaci algorytmów, ale uczeń powinien poznać i umieć je zastosować w konkretnych sytuacjach znanych i nowych [7].

2.2.3. Zapisywanie algorytmów w postaci procedur.

Ponieważ programowanie nie występuje w programie informatyki dla gimnazjum, „procedury dla komputera” należy rozumieć tutaj bardzo szeroko. Mogą to być gotowe programy, a uczeń może jedynie zmieniać wartości parametrów, by badać ich wpływ na postać rozwiązania, lub schematy algorytmów, takie jak np. w ELI, których tworzenie nie wymaga znajomości programowania, a może być wykonane na podstawie słownego opisu algorytmów. Celem tego etapu zajęć powinno być wydzielanie z opisów rozwiązań podstawowych konstrukcji algorytmicznych, np. w postaci „podalgorytmów”, które mogą być wykorzystane w rozwiązaniach innych problemów, a którym w realizacjach komputerowych odpowiadają podprogramy (procedury).

2.2.4. Algorytmy rekurencyjne.

Rekurencja, jako sposób postępowania, a dokładniej - sposób zapisywania pewnych działań, takich, jak klasyczne: jedzenie kaszki, tańczenie czy śpiewanie, może pojawić się w nauczaniu dość wcześnie [4]. Pomocne w tym może być posłużenie się językiem np. Logo. Rekurencyjny opis postępowania w dość prosty sposób wiedzie ku rekurencyjnej metodzie rozwiązania danej sytuacji problemowej np. przestawiania krążków między palikami (tzw. Wieże Hanoi)¹ oraz przy poszukiwaniu wyjścia z labiryntu (metodą „idź przed siebie”). Sam zapis rekurencyjny należy traktować, jako jeden z możliwych sposobów opisu, ale nie jedyny. Większość bowiem algorytmów rekurencyjnych ma swoje wersje iteracyjne i należy również przeznaczyć fragment lekcji na dyskusję wokół tematu „rekurencja a iteracja”.

Zadanie Fibonacciego o królikach² jest najprostszym problemem, w rozwiązaniu którego można posłużyć się rekurencją i które może być użyte do wyjaśnienia niemal wszystkich aspektów rekurencji. I może właśnie prostota tego problemu powoduje, że

¹ Łamigłóвка polega na przeniesieniu pewnej liczby krążków różnej wielkości z otworami z palika A na palik B, z możliwością posłużenia się przy tym palikiem C, w taki sposób, że pojedynczy ruch polega na przeniesieniu jednego krążka między dwoma palikami; w żadnej chwili rozwiązywania łamigłóвки, większy krążek nie może leżeć na mniejszym.

² Oryginalne pytanie Fibonacciego brzmiało: ile par królików będzie po upływie roku?.

nie przykuwa on specjalnie uwagi uczniów. Ponadto, jest to również znakomita okazja do zaprojektowania zajęć interdyscyplinarnych, obejmujących niemal wszystkie bloki nauczania w szkole podstawowej, gimnazjum.

2.2.5. Rozwiązywanie zadań metodą zstępującą.

Większość przytoczonych wyżej sytuacji problemowych ma dość złożony charakter i ich rozwiązanie, w wyniku dyskusji z uczniami, powinno polegać na rozbiciu problemu na mniejsze części. Jest to nic innego, jak metoda zstępująca. To podejście umożliwia m.in. wydzielenie w problemach podobnych fragmentów i tworzenie z nich podproblemów, do rozwiązania których można stosować „podprogramy”. W szczególnych przypadkach, podproblemy mogą być znanymi już uczniom problemami, dla których znają rozwiązania. W konsekwencji, w naturalny sposób może pojawić się metoda rozwiązywania „dziel-i-zwyciężaj”, a wraz z nią - rekurencja i programowanie dynamiczne³ [8].

2.2.6. Testowanie i ocena algorytmów. Własności algorytmów.

Można się w tym posłużyć własnymi programami lub projektami algorytmów, zapisanymi np. w programie ELI. Testowanie i ocena algorytmów, poza sprawdzeniem konkretnych algorytmów, powinny mieć również na celu ich zilustrowanie. Komputery nie zawsze są w stanie nam pomóc w rozwiązywaniu rzeczywistych problemów, a dotychczasowe doświadczenia wskazują, że ich moc obliczeniowa w dużym stopniu jest uzależniona od użytego w rozwiązaniu algorytmu.

Najważniejsze własności algorytmów można podzielić na trzy grupy:

1. poprawność – poprawność algorytmu określamy względem specyfikacji⁴ problemu, to znaczy że dla każdego poprawnych danych otrzymamy odpowiedni wynik. Algorytm można uznać że jest poprawny, gdy daje rozwiązanie w skończonej liczbie kroków, czyli gdy jest skuteczny.
2. skończoność – każde działanie programu musi być skończone,
3. efektywność – mała złożoność obliczeniowa, gwarantująca jego użyteczność w praktycznych sytuacjach.

³ W znaczeniu stosowanym tutaj: działania mającym na celu podjęcie decyzji dającej jak najlepsze efekty np. poszukiwanie wyjścia z labiryntu, pakowania plecaka, wydawanie reszty .

⁴ Ścisła definicja problemu, składa się z danych, wyników oraz związków zachodzących między nimi.

2.3. Znaczenie algorytmów w programach nauczania.

Algorytmy, jako opisy postępowań prowadzących do wyznaczonego celu lub będących opisem rozwiązania postawionego problemu, pojawiają się w wielu dziedzinach nauczania. Występują również wokół nas, w wielu codziennych sytuacjach. Takie problemy, jak: porządkowanie liczb, słów i dat, szukanie drogi wyjścia z labiryntu, czy optymalne pakowanie plecaka, można rozważać na wielu różnych lekcjach, np. matematyki, informatyki, fizyki i techniki. Celem zajmowania się tymi problemami jest wykształcenie u uczniów pewnych strategii postępowania w sytuacjach podobnych do tych omawianych lub odmiennych, ale wymagających podjęcia decyzji w konkretnej sytuacji praktycznej.

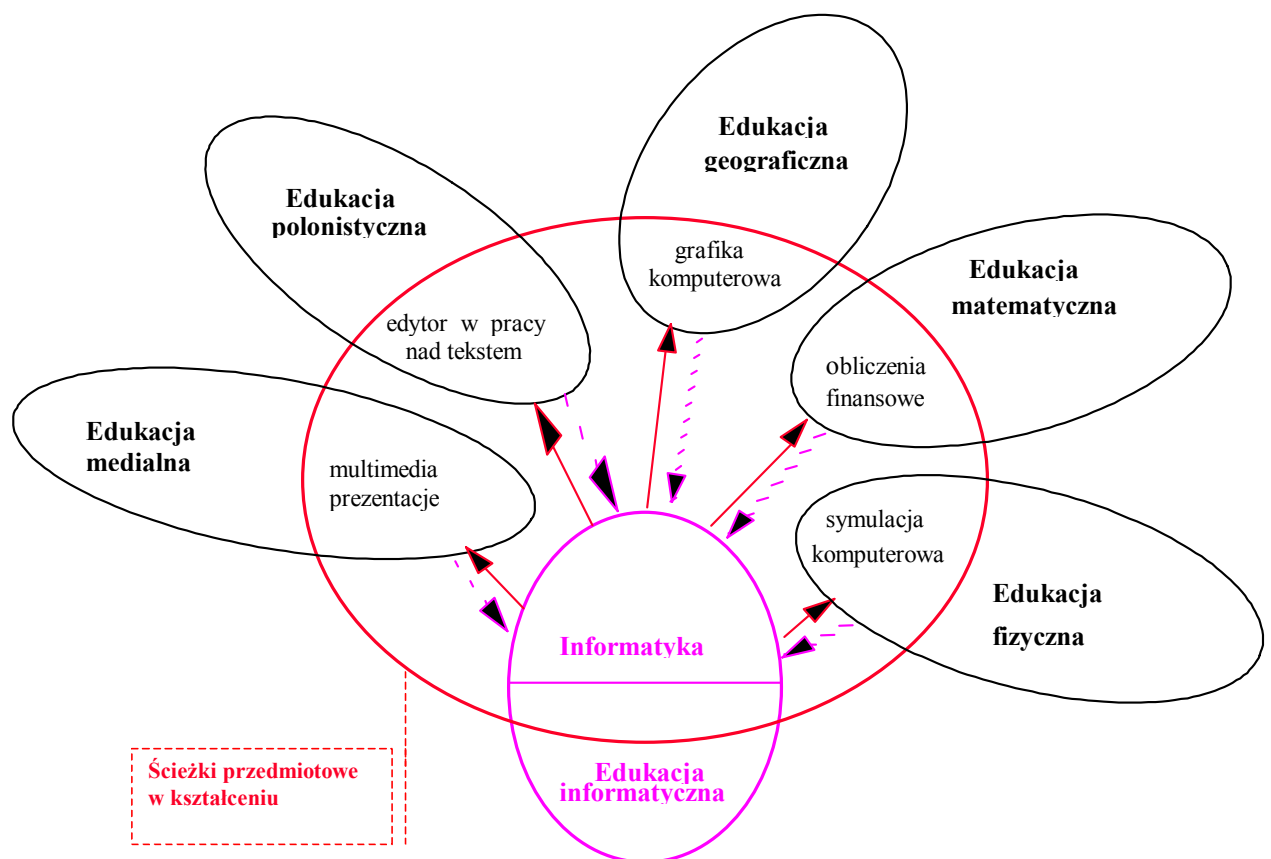
W dalszej części pracy, wykorzystując wcześniej pokazane elementy, zaproponuję przebieg zajęć. Posłużę się tutaj algorytmem zachłannym; jako przykład posłużę problem wydawania reszty.

2.4. Ścieżki przedmiotowe

Technologia informacyjna wzbogaca istniejące, a również stwarza nowe warunki do rozwijania umiejętności uczniów w większości zakresów. Trudno dzisiaj mówić o uczeniu jednego przedmiotu nie wykorzystując pojęć zagadnień z innych dziedzin. To „łączenie” wiedzy szczególną rolę odgrywa w ogólnym rozwoju ucznia. W przypadku:

- uczenia się - wzbogaca sposoby i wspomaga możliwości zdobywania wiedzy i umiejętności, ułatwia rozwiązywanie problemów i podejmowanie decyzji, usprawnia proces uczenia się oraz integruje doświadczenia i różne elementy wiedzy;
- myślenia - wspomaga radzenie sobie z niepewnością i złożonością zjawisk oraz umożliwia ich całościowe postrzeganie, wspomaga pracę twórczą;
- poszukiwania - jest technologią, służącą do poszukiwania, porządkowania i wykorzystywania informacji pochodzących z różnych źródeł;
- działania - usprawnia organizację pracy i ułatwia posługiwanie się wieloma technikami i narzędziami pracy, usprawnia projektowanie działań;

- doskonalenia się - przez swój ciągły rozwój, technologia informacyjna jest przygotowana do zmian i może przyczynić się do kształtowania postawy elastycznej i uwzględniającej zmiany, wyrabia przez to potrzebę ciągłego uczenia się;
- komunikowania się - dostarcza środków do indywidualnych i grupowych prezentacji oraz do skutecznego komunikowania, uczy więc porozumiewania się oraz sprzyja kontaktom międzyludzkim bez barier i uprzedzeń;
- współpracy - ułatwia pracę w grupie (lokalnej i globalnej) i osiągnięcie porozumienia, może być wykorzystana do nawiązania i podtrzymywania kontaktów oraz budowania więzi międzyludzkich.



Rys. 1.6 Przykładowa ścieżka przedmiotowa – edukacja informatyczna w kształceniu ogólnym.

3. Problem reszty -przykład algorytmu zachłannego.

3.1. Algorytmy zachłanne.

Algorytm zachłanny (*ang. greedy algorithm*) wykonuje zawsze działanie, które wydaje się w danej chwili najkorzystniejsze. Wybiera zatem lokalnie optymalną możliwość w nadziei, że doprowadzi ona do globalnie optymalnego rozwiązania.

Metoda zachłanna polega na konstrukcji rozwiązania problemu w następujący sposób.

- rozpoczynamy konstrukcję, rozwiązania od rozwiązania pustego,
- następnie spośród dostępnych elementów (dróg dalszej konstrukcji rozwiązania) wybieramy ten, który w danej chwili wydaje się, być najlepszy,
- odrzucamy elementy, które nie dają się, dopasować do już skonstruowanego rozwiązania, (będą, zbędne w dalszych rozważaniach),
- kontynuujemy takie postępowanie, aż do osiągnięcia pełnego rozwiązania.

Zauważmy, że w algorytmie zachłannym nigdy nie poprawiamy już skonstruowanego rozwiązania i nie wycofujemy się z raz dokonanych wyborów. W algorytmie zachłannym bardzo ważne i często trudne jest określenie kryterium wyboru elementu aktualnie najlepszego. Następną istotną sprawą jest udowodnienie, że proponowana metoda zachłanna prowadzi do skonstruowania rozwiązania najbardziej optymalnego.

3.2. Problem reszty – jako przykładowy konspekt lekcji.

3.2.1. Cele kształcenia.

Uczeń potrafi (wymagane przygotowanie ucznia) :

- posługiwać się językiem technologii informatycznej,
- formułować proste sytuacje problemowe,
- podać proste problemy dotyczące optymalizacji rozwiązań z różnych dziedzin,
- porządkować i klasyfikować podane wiadomości i informacje, właściwie je przeanalizować i pogrupować,
- współpracować w grupie przy opracowywaniu rozwiązania problemu, formułować oraz przedstawiać swoje rozwiązania,
- być odpowiedzialny wobec stawianych zadań [9].

3.2.2. Treści problemowe.

Sytuacje problemowe, które będziemy stwarzać uczniom w szkole, powinny :

- *motywować ich*, do zajmowania się nimi,
- zawierać w sobie *element zaciekawienia*, będącego zaprzeczeniem rutyny i stanowiącego wyzwanie „*jak to rozwiązać ?*”

3.2.3. Rozwiązywanie problemów i podejmowanie decyzji.

- Przykłady tworzenia i badania prostych modeli o niewielkiej liczbie parametrów.
- Algorytmy wokół nas, przykłady algorytmów rozwiązywania problemów praktycznych i szkolnych.
- Współdziałanie w grupie, z uwzględnieniem podziału zadań, w pracy nad wspólnymi projektami realizowanymi z wykorzystaniem TI [9].

3.2.4. Działania.

- Uczniowie poznają sposoby porządkowania informacji.
- Uczniowie poznają różne formy organizacji danych. Uczniowie wyszukują i zapisują informacje w bazach danych.
- Uczniowie planują, rozwijają, testują i modyfikują ciągi instrukcji i procedur do kontrolowania urządzeń, np. zółwia Logo.
- Uczniowie poznają przykłady algorytmów rozwiązywania problemów praktycznych i szkolnych.
- Uczniowie ściśle formułują przykładowe sytuacje problemowe. Opisują algorytmy w języku potocznym.
- Uczniowie poznają i stosują struktury języków opisu algorytmów.
- Uczniowie zapisują algorytmy w postaci procedur i wykonują je na komputerze.
- Uczniowie poznają przykłady algorytmów rekurencyjnych.
- Uczniowie zapoznają się z działaniem wybranych algorytmów, rozwiązujących proste problemy. Testują i oceniają algorytmy [9].

3.2.5. Stawiamy problem:

Zachłanny algorytm wydawania reszty¹

Dane: Nominały monet oraz reszta do wydania.

Wynik: Przedstawienie reszty w postaci najmniejszej ilości monet.

Krok iteracyjny: Dopóki reszta nie jest jeszcze równa zero, odejmij od niej największy możliwy nominal

Przykład

Kasjer ma wydać resztę, będącą dowolną kwotą przy użyciu minimalnej liczby monet. mieszczącej się w niej nominal i wydać odpowiednią monetę.

Rozwiązanie oparte na algorytmie zachłannym

Najpierw używamy monety o największej dopuszczalnej wartości, redukując w ten sposób problem do wypłacenia mniejszej ilości monet.

Rozwiązanie:

Aby wydać 0,99 PLN, kasjer wypłaci:

- pięćdziesiąt groszy (zostawiając do wypłacenia 0,49 PLN), następnie
- dwie dwudziestogroszówki (zostaje 0,09 PLN),
- pięć groszy (zostaje 0,04 PLN), w końcu,
- dwie monety dwugroszowe.

Problem reszty - wybór metody rozwiązania postawionego problemu.

Proponujemy rozwiązanie problemu w formie zabawy wyobraźni i pomysłowości z wykorzystaniem metody graficznej. Obie grupy rozwiązanie proponują na przygotowanych rysunkach; wpisują swoje oczekiwania, propozycje, rozwiązania itp.

⇒ *dzielimy klasę na dwie grupy; jedna grupa będzie odgrywała rolę sprzedawcy, druga zaś rolę kupującego,*

⇒ *dla obu grup zadajemy pytanie wprowadzające do dyskusji :*

- *sprzedawca: „Jakimi monetami chciałbyś wydać resztę?” (rys. Sprzedawcy)*

¹ Problem reszty, jak każda w nim moneta ma dwie strony: odbierający resztę chciałby dostać jak najmniej monet, a wydający – pozbyć się ich jak najwięcej. Obie tendencje mają swoje zachłanne realizacje, ale decyzja jest jednak w rękach wydającego. Może on wydawać nam jak najwięcej małych monet; w tym przypadku postępując zgodnie z algorytmem, mógłby postępować z oczekiwaniami klientów, a przy tym samym miałby mniej do roboty

- *kupujący* – „**W jakich monetach chciałbyś otrzymać resztę ?**” (rys. *Kupujący*).

Sposób przeprowadzenia zabawy wprowadzającej uczniów w zadany problem :

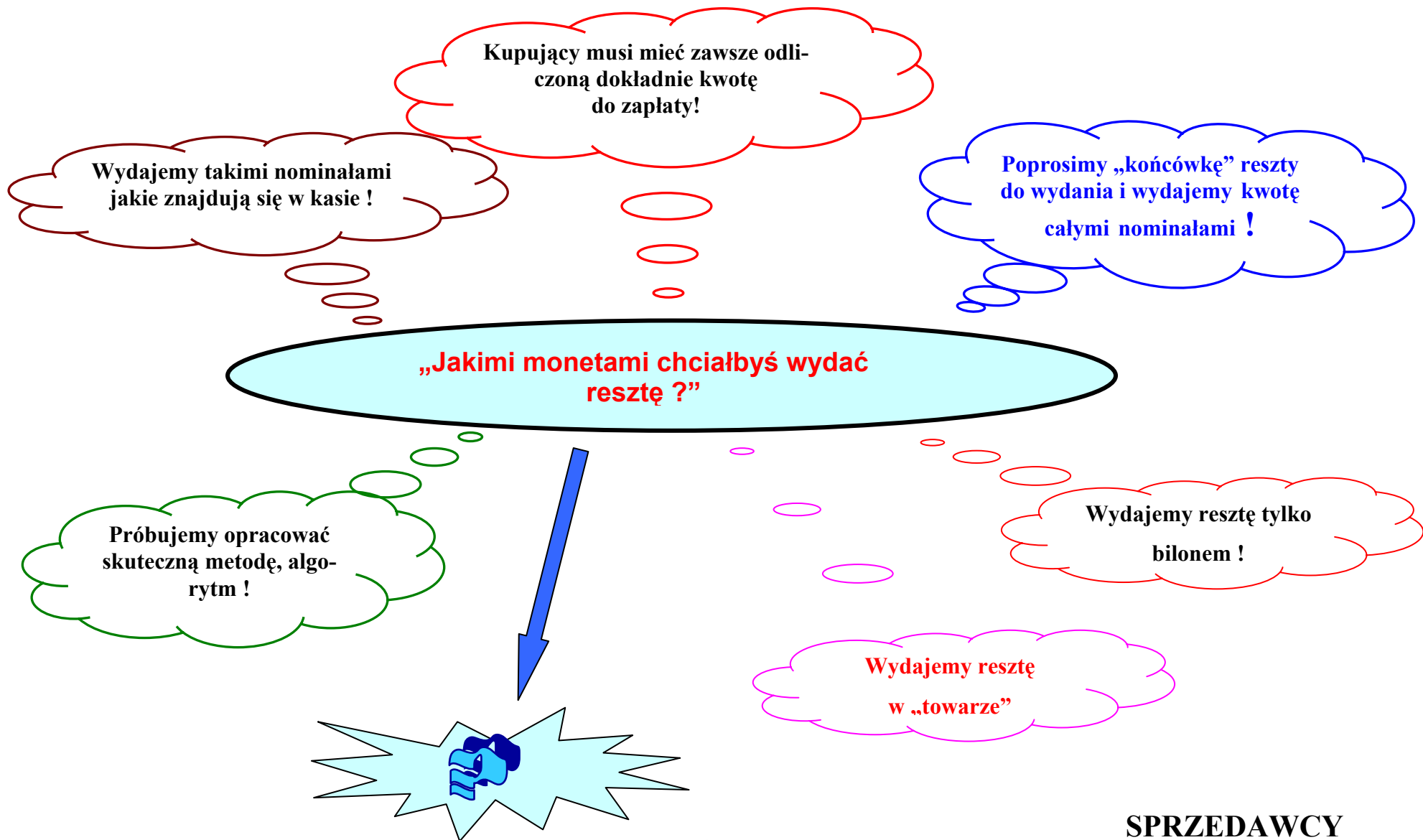
⇒ *dzielimy klasę na kilka zespołów (bloków) po kilku uczniów;*

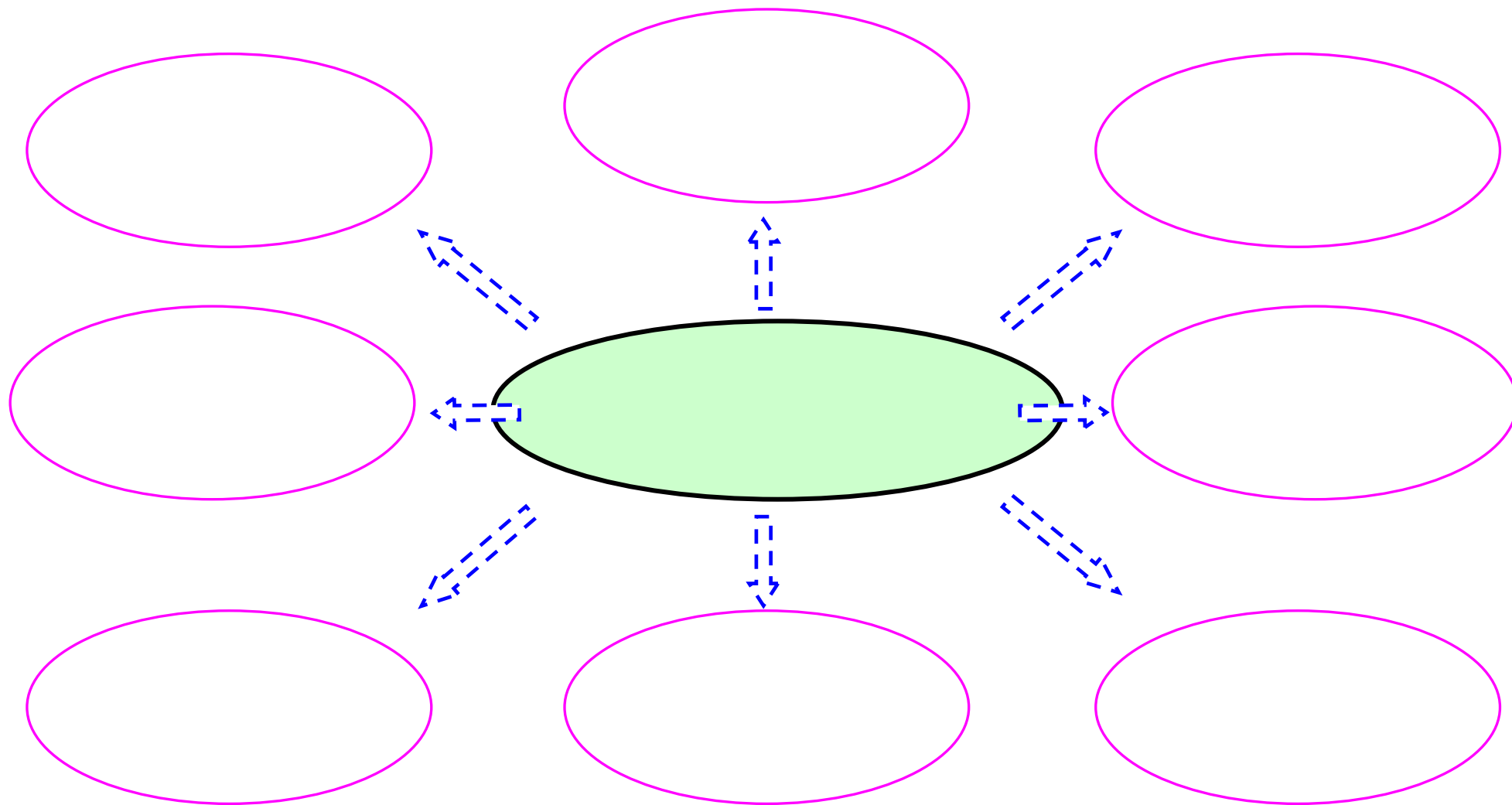
- ▶ *pierwszy zespół: - uczeń pobierający nominały monet - nominały monet powinny być przygotowane wcześniej na odpowiednich kartonikach oraz uczniowie odpowiedzialni za resztę wpisywaną każdorazowo na kartoniku,*
- ▶ *drugi zespół: – uczniowie (po 3 osoby dla każdego bloku działań), którzy każdorazowo będą wykonywać zadane działania matematyczne,*
- ▶ *trzeci zespół: uczniowie (po 3 osoby dla każdego bloku decyzyjnego), którzy każdorazowo będą sprawdzać zadany warunek matematyczny i decydować o dalszym sposobie postępowania,*
- ▶ *czwarty zespół : zespół zbierający wyniki działań,*

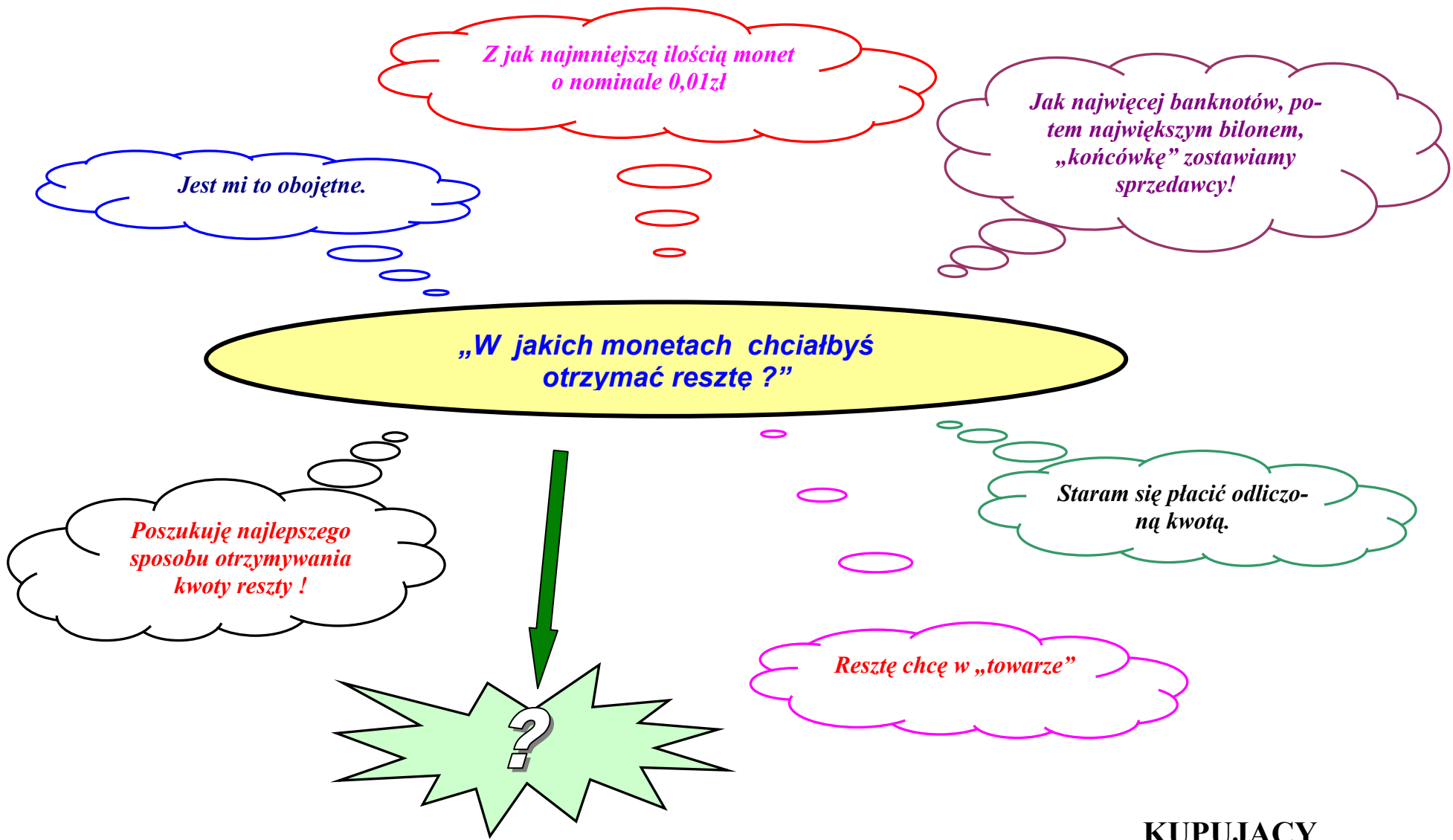
⇒ *ustalamy dla każdego zespołu kierunki działań : **skąd (od kogo) pobiera dane – dokąd (komu) przekazuje wyniki swoich działań,***

⇒ *ustalamy sposób postępowania w formie: **pobierz dane – sprawdź postawiony warunek – wykonaj działanie zgodnie z zadany warunkiem,***

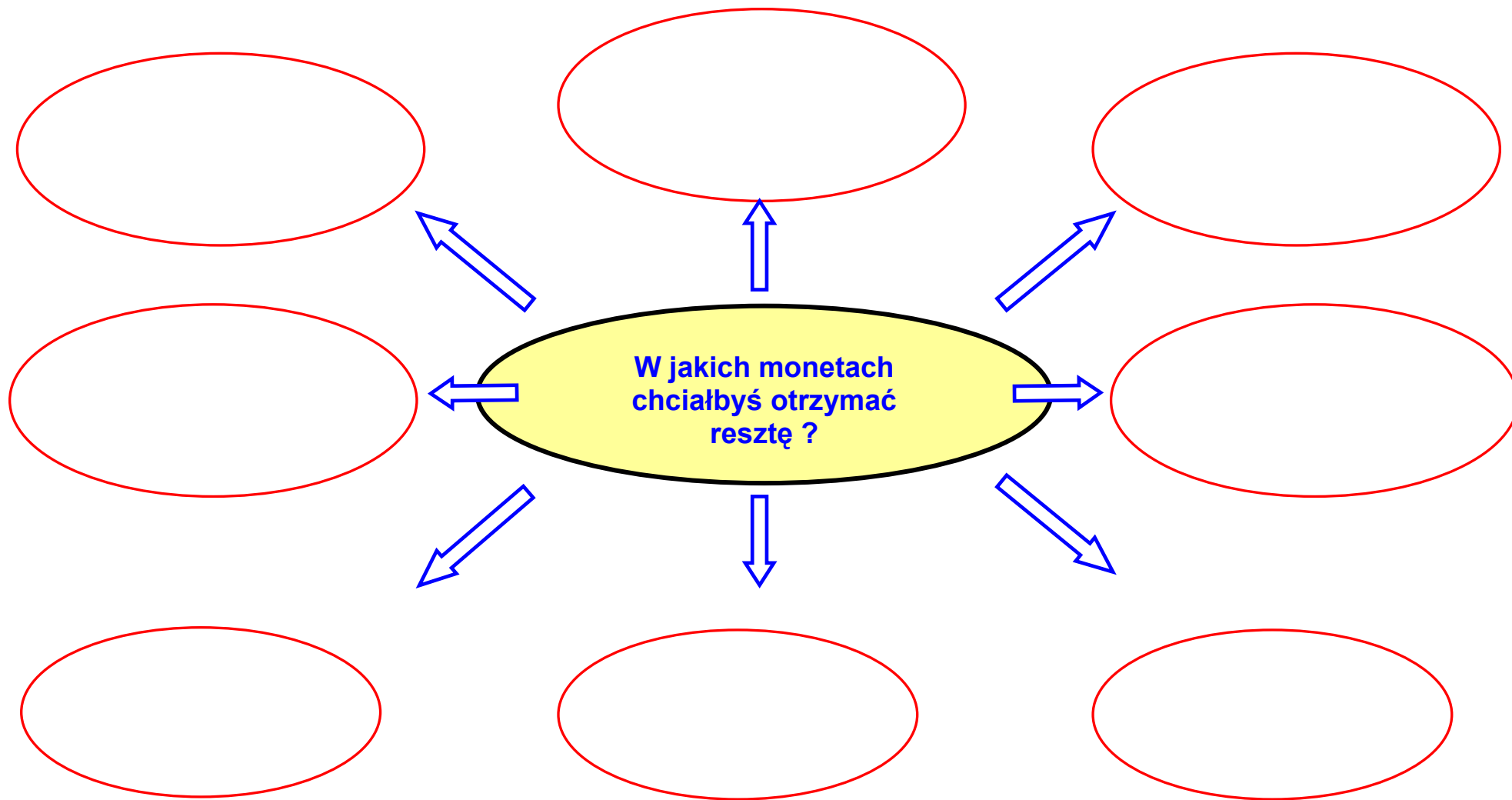
⇒ *każdy krok postępowania (z pomocą nauczyciela) będzie rysowany na tablicy w postaci umownego schematu blokowego.*







KUPUJĄCY



3.2.6. Dyskusja nad wyborem metody rozwiązania .

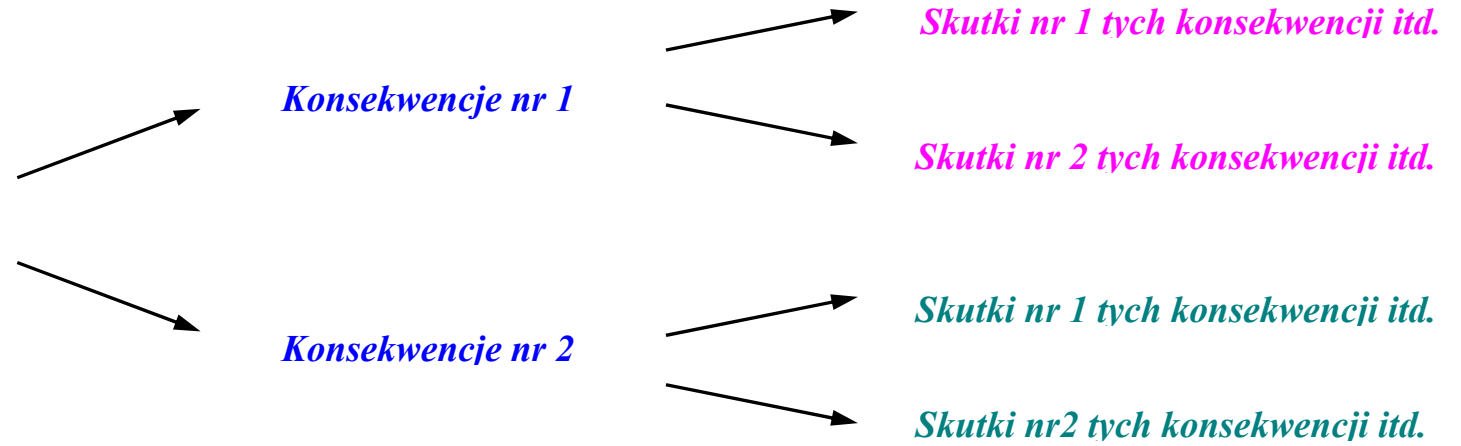
W dyskusji oczekujemy uzasadnienia proponowanych rozwiązań, kierujemy się tutaj zasadami:

- **każdy ma prawo podać swój pomysł,**
- **nie wolno krytykować pomysłów innych,**
- **wszystkie rozwiązania są zapisywane.**

Celem dyskusji jest wybór najlepszej metody postępowania przy wydawaniu reszty. Wspólnie z nauczycielem (jego rola ogranicza się do umiejętnego naprowadzania uczniów przy wyborze metody) uczniowie analizują wszystkie zaproponowane rozwiązania (za i przeciw) i wybierają to, które według nich jest właściwe (dla sprzedawcy i kupującego).

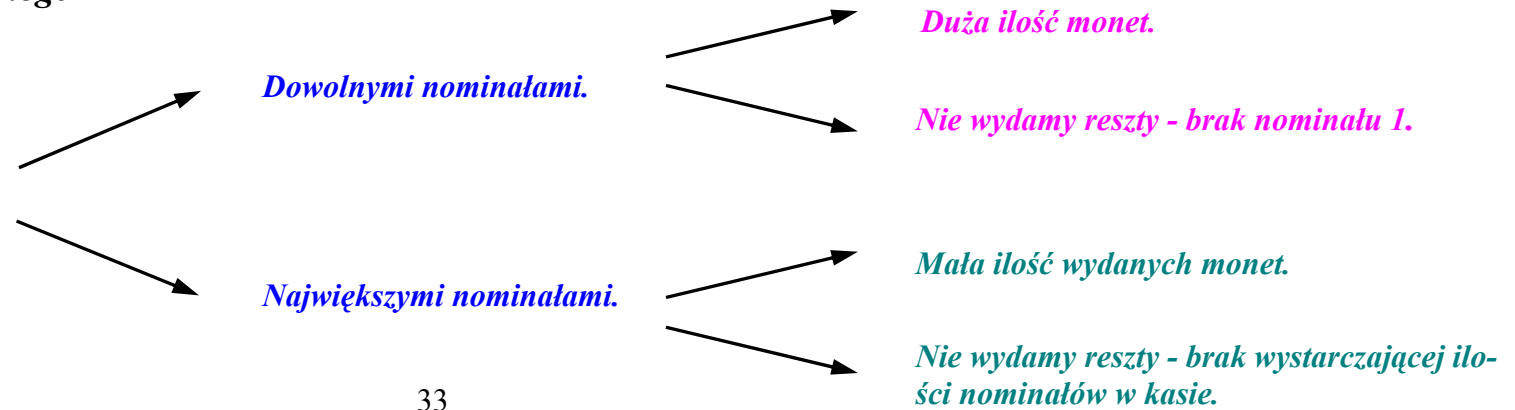
Dla lepszej organizacji i prezentacji argumentów i wynikających z ich przyjęcia konsekwencji, możemy wykorzystać tzw. „drzewko decyzyjne”. Pozwoli ono lepiej prześledzić proces myślenia uczniów przy wyborze dalszej metody postępowania. Powyższe „drzewko” możemy w trakcie dyskusji całej klasy, rysować na tablicy. Przykład „drzewka decyzyjnego przedstawiono poniżej:

ROZWIĄZANIE NR 1



Przykład „drzewka decyzyjnego”

**Wydajemy resztę tylko bi-
lonem !**



3.2.7. Przykłady do zadań. Wnioski z nich wypływające.

Zadanie 1. Sformułuj specyfikację problemu wydawania reszty w polskiej walucie tj. określ dane, wyniki oraz warunki, jakie wiążą wyniki z danymi. Podaj przykłady wydawania reszty¹.

Przykłady do omówienia:

Lp.	Kwota do wydania	Złote	Grosze
1.	83,76 PLN	50+20+10+2+1	50+20+5+1
2.	49,99 PLN	20+20+5+2+2	50+20+20+5+2+2

3.	31,23	a)	20+10+1	20+2+1
		b)	10+10+10+1	10+10+2+1
		c)	10+10+5+5+1	10+10+1+1+1
		d)	10+5+5+5+2+2+2	10+5+5+2+1

WNIOSKI:

- obecność nominału 1 przy wydawaniu reszty jest niezbędna,
- w kasie są wszystkie nominały monet w wystarczającej ilości,
- sumę kilka monet o mniejszym nominale możemy zamienić na jedną o wyższym nominale,
- każdą kwotę reszty możemy wydać nominałem 1.

Zadanie 2. Podaj przykłady na to, że jeśli wśród nominałów nie ma 1, to problem reszty może nie mieć rozwiązania.

Przykłady do omówienia:

Lp.	Kwota do wydania zł	Złote	Grosze
1.	83,78	50+20+10+2+1	50+20+5+2 + ... brak nominału 1
2.	31,18	20+10+... brak nominału 1	10+5+2 +... brak nominału 1

¹ Nominały większości walut nie są przypadkowymi liczbami, zostały one tak dobrane, by sprzedawcy dysponujący dostatecznym zasobem monet mieli prosty sposób wydawania reszty za pomocą najmniejszej ilości monet. W rezultacie, sami mają mniej okazji, by się pomylić a klienci mają mniej monet do noszenia.

WNIOSKI:

- nie ma na świecie waluty, która nie zawierałaby monety o nominale 1,
- nominały monet nie są przypadkowymi wartościami,

3.2.8. Zachłanna metoda postępowania przy wydawaniu reszty.

Omawiając metodę „postępowania zachłannego” przy wydawaniu reszty musimy odpowiedzieć na pytanie :

Jak wydać zadaną kwotę reszty najmniejszą ilością monet?

- jeżeli w zadaniu, które mamy zrealizować, można wydzielić kolejne etapy, to działanie polegające na tym, że w każdym kroku staramy się postępować jak najszybciej, zyskać jak najwięcej i stracić jak najmniej, nazwiemy postępowaniem zachłannym,
- w każdym kroku działania chcemy wykonać najlepszy w naszej ocenie ruch,
- operacja zamiany większej liczby monet o małych nominałach na monetę o większym nominale sugeruje postępowanie zachłanne, w którym od razu staramy się używać jak największych nominałów.

3.2.9. Zachłanna strategia postępowania - algorytm postępowania.

W celu lepszego zrozumienia idei algorytmu, proponuję zabawę z uczniami w oparciu o przedstawiony schemat algorytmu. Dopiero po przeprowadzonej zabawie i dyskusji, można przystąpić do rysowania schematu blokowego algorytmu oraz do sprawdzenia jego działania na innych „resztach do wydania”.

a) algorytm blokowy:

- opis krok po kroku rozwiązania postawionego problemu lub sposobu osiągnięcia jakiegoś celu, który powinien być dokładnie określony w postaci umownych bloków decyzyjnych.

b) cechy algorytmu:

- dane winny być dokładnie sprecyzowane (winny być uniwersalne),

- cel (wynik), który chcemy osiągnąć jasno określony oraz powiązany z danymi; to powiązanie danych z wynikiem – czyli jak rodzaj i ilość oraz jakość danych wpłynie na wynik (nazywamy to specyfikacją problemu²).

3.2.10. Algorytm wydawania reszty.

Przyjmujemy założenia :

R – reszta do wydania,

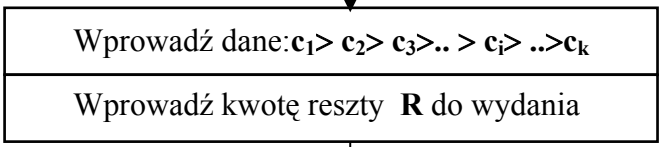
c_i - nominały monet, którymi będzie wydawana reszta.

Rozwiązanie :

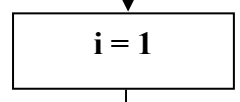
<i>Dane :</i>	Nominały monet, którymi dysponujesz oraz reszta R do wydania.
<i>Wynik :</i>	Przedstawienie kwoty do wydania (reszty) w postaci najmniejszej liczby monet.
<i>Krok 1:</i>	Wybierz nominał c_i monety ze zbioru monet, którymi aktualnie dysponujesz.
<i>Krok 2:</i>	Odejmij go od kwoty reszty, którą masz wydać tj. $R = R - c_i$
<i>Krok 3:</i>	Jeżeli wynik odejmowania jest $R - c_i \geq 0$; - wydaj odejmowany nominał c_i , i przejdź do <i>kroku 4</i> . W przeciwnym razie pobierz następny nominał (<i>krok 1</i>) i przejdź do <i>kroku 2</i> .
<i>Krok 4:</i>	Jeżeli $R = 0$; to zakończyłeś wydawanie reszty, w przeciwnym razie wróć do <i>kroku 2</i> .

² Ścisła definicja problemu. Patrz przypisy wcześniejsze.

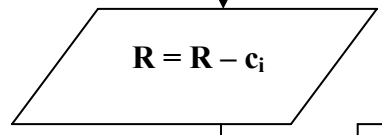
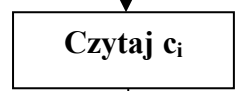
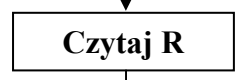
DANE



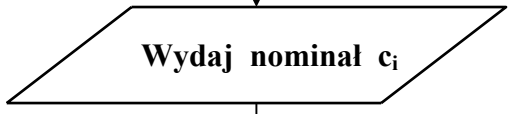
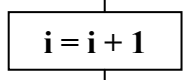
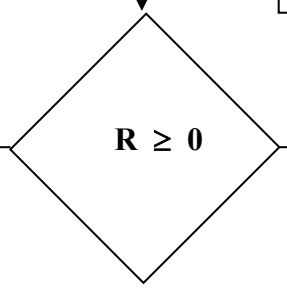
KROK 1



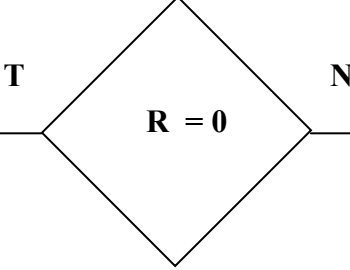
KROK 2



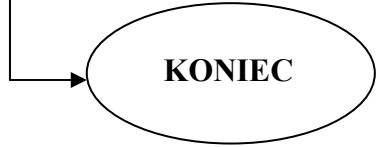
KROK 3



KROK 4



WYNIK



Sprawdzenie działania algorytmu :

<u>Dane :</u>	Nominały monet: 200; 100; 50; 20; 10; 5; 2; 1; 0,50; 0,20; 0,10; 0,05; 0,02; 0,01
	Reszta do wydania: 31,23
<u>Wynik :</u>	Przedstawienie kwoty do wydania (reszty) w postaci najmniejszej liczby monet.

Etap	Pobra- ny nomi- nał c_i	Reszta R do wydania	Działanie	Rezultat działania	Wykonaj i wróć do <i>(Kroku ...)</i>
1	200 zł	31,23 zł	31,23 - 200	< 0	Pomiń nominał 200 zł (1)
2	100 zł	31,23 zł	31,23 - 100	< 0	Pomiń nominał 100 zł (1)
3	50 zł	31,23 zł	31,23 - 50	< 0	Pomiń nominał 50 zł (1)
4	20 zł	31,23 zł	31,23 - 20	> 0	Wydadaj nominał 20 zł (2)
5		11,23 zł			
6	20 zł	11,23 zł	11,23 - 20	< 0	Pomiń nominał 20 zł (1)
7	10 zł	11,23 zł	11,23 - 10	> 0	Wydadaj nominał 10 zł (2)
8		1,23 zł			
9	10 zł	1,23 zł	1,23 - 10	< 0	Pomiń nominał 10 zł (1)
10	5 zł	1,23 zł	1,23 - 5	< 0	Pomiń nominał 5 zł (1)
11	2 zł	1,23 zł	1,23 - 2	< 0	Pomiń nominał 2 zł (1)
12	1 zł	1,23 zł	1,23 - 1	> 0	Wydadaj nominał 1 zł (2)
13		0,23 zł			
14	1 zł	0,23 zł	0,23 - 1	< 0	Pomiń nominał 1 zł (1)
15	0,50 zł	0,23 zł	0,23 - 0,50	< 0	Pomiń nominał 0,50 zł (1)
16	0,20 zł	0,23 zł	0,23 - 0,20	> 0	Wydadaj nominał 0,20 zł (2)
17		0,03 zł			
18	0,20 zł	0,03 zł	0,03 - 0,20	< 0	Pomiń nominał 0,20 zł (1)
19	0,10 zł	0,03 zł	0,03 - 0,10	< 0	Pomiń nominał 0,10 zł (1)
20	0,05 zł	0,03 zł	0,03 - 0,05	< 0	Pomiń nominał 0,05 zł (1)
21	0,02 zł	0,03 zł	0,03 - 0,02	> 0	Wydadaj nominał 0,02 zł (2)
22		0,01 zł			
23	0,02 zł	0,01 zł	0,01 - 0,02	< 0	Pomiń nominał 0,02 zł (1)
24	0,01 zł	0,01 zł	0,01 - 0,01	= 0	Wydadaj nominał 0,01 zł

WYDANE NOMINAŁY MONET : 20zł; 10zł; 1zł; 0,20zł; ,0,02zł; 0,01zł

3.2.11. Zadania do wykonania przez uczniów w formie ćwiczeń.

Zadanie 3. Zastosuj zachłanny algorytm wydawania reszty do utworzenia kwot groszowych : 63, 87, 117 z możliwie najmniejszej liczby monet. Sprawdź na tych przykładach, czy czasem nie można utworzyć tych reszt z jeszcze mniejszej liczby monet. Czy zachłanny algorytm wydawania reszty za pomocą polskich monet powiększonych o nominał 21 groszy gwarantuje, że każdą resztę otrzymamy w postaci najmniejszej liczby monet ? Rozważ kwoty 25 i 63.

Zadanie 4. Przetestuj działanie algorytmu na dowolnym przykładzie liczbowym. Sprawdź czy będzie on poprawny w przypadku, kiedy w zbiorze nominałów do wydania nie będzie jednego lub kilku nominałów. Czy wydana wówczas kwota reszty będzie spełniała cechy działania zachłannego ?

3.2.12. Osiągnięcia.

Uczeń:

- a) potrafi współpracować w grupie i podejmować decyzje przy realizacji wspólnego zadania,
- b) potrafi przekazywać swoje pomysły i informacje w różnych formach, używając tekstu, rysunku,
- c) potrafi prowadzić obserwacje i posługiwać się modelami przy objaśnianiu zagadnień i problemów,
- d) potrafi samodzielnie zapisywać proste algorytmy w postaci procedur,
- e) zna proste sposoby i przykłady testowania zaproponowanych rozwiązań,
- f) potrafi zaproponować rozwiązania alternatywne,
- g) zaproponowane rozwiązania potrafi przetwarzać, modyfikować wykorzystywać w innych zadaniach,
- h) posiada umiejętność dyskusowania i wprowadzania zmian [9].

Umiejętności, czyli co uczeń będzie umiał po zakończeniu kształcenia:

- a) samodzielnie i bezpiecznie posługiwać się systemem komputerowym i jego oprogramowaniem,

- b) opisywać czynności wykonywane przy korzystaniu z komputera, stosując poprawną terminologię,
- c) wybierać, łączyć i celowo stosować różne narzędzia informatyczne do rozwiązywania typowych problemów praktycznych i szkolnych,
- d) korzystać z tekstów i dokumentacji technicznej w postaci programów,
- e) opisywać i stosować proste algorytmy,
- f) podać przykłady algorytmów rozwiązywania problemów praktycznych i szkolnych,
- g) korzystać z różnych multimedialnych i rozproszonych źródeł informacji dostępnych za pomocą komputera, modelować problemy [9].

3.3. Problem reszty – program zapisany w języku Pascal.

3.3.1. Opis zadania.

Dane: zbiór różnych liczb naturalnych c_1, c_2, \dots, c_n (przyjmijmy, że $c_1 < c_2 < \dots < c_n$) nazywanych nominałami monet oraz liczba naturalna R nazywana resztą,;

Zadanie: wydać resztę R używając jak najmniejszej liczby monet, tzn. znaleźć liczby naturalne d_1, d_2, \dots, d_n takie, by $R = \sum_{i=1}^n d_i c_i$ oraz by liczba $\sum_{i=1}^n d_i$ była jak najmniejsza.

3.3.2. Zapis w programie Pascal – wydawanie reszty.

```
program reszta;
uses Crt;
var aNominaly : array[1..14] of Integer;
iReszta, iWynik : Integer;
i : Integer;
begin
    aNominaly[1]:=1;
    aNominaly[2]:=2;
    aNominaly[3]:=5;
    aNominaly[4]:=10;
    aNominaly[5]:=20;
    aNominaly[6]:=50;
    aNominaly[7]:=100;
    aNominaly[8]:=200;
    aNominaly[9]:=500;
    aNominaly[10]:=1000;
    aNominaly[11]:=2000;
    aNominaly[12]:=5000;
    aNominaly[13]:=10000;
    aNominaly[14]:=20000;
    ClrScr;
    Write('Podaj reszta: ');
    ReadLn(iReszta);
    while iReszta>0 do
    begin
        for i:=14 downto 1 do
        begin
            if iReszta>=aNominaly[i] then
            begin
                WriteLn('Wydac->',aNominaly[i]);
                iReszta:=iReszta-aNominaly[i];
                i:=14;
            end;
        end;
    end;
end.
```

Wnioski.

1. Dlaczego algorytmy są ważne? Czy algorytmy są niezbędnym elementem kształcenia? Postaram się uzasadnić tutaj pogląd, że są to jedne z najważniejszych obiektów, jakie powinny pojawić się w trakcie uczenia się i nauczania. Nie chodzi o nauczanie samych algorytmów jako gotowych przepisów postępowania lub programów dla komputerów. Nauczaniem o algorytmach powinny rządzić ogólniejsze zasady, którym jest podporządkowana cała edukacja. Obecna reforma szkolnictwa w Polsce ma na celu m.in. zbliżenie szkoły do indywidualnych oczekiwań oraz możliwości uczniów. W tym celu szkoła powinna umożliwiać i podsycać działania związane z nieskrępowaną twórczością uczniów. Znakomicie poddaje się temu cała dziedzina związana z rozwiązywaniem problemów, do której można zaliczyć również tę część algorytmiki, która jest rozumiana jako poszukiwanie rozwiązań problemów w postaci algorytmów. O algorytmach nie można uczyć w sposób statyczny, zamknięty, czy programowany, chociaż najczęściej docelowa postać algorytmu ma postać ścisłego opisu (np. jako program) kolejnych działań (kroków), które mają być wykonane. Algorytm nie powinien być podawany jako gotowa odpowiedź, będąca rozwiązaniem rozważanego problemu, gdyż jednym z głównych celów nauki rozwiązywania problemów jest poznanie i przyswojenie sobie przez uczącego się ogólnej wiedzy o rozwiązywaniu problemów i tworzeniu rozwiązań, użytecznej przede wszystkim w sytuacjach, gdy pojawiają się nowe problemy.
2. Zatem tym, co ma pozostawać po nauce o algorytmach, powinny być nie same przepisy postępowania i ba nawet nie opisy ogólnych technik rozwiązywania, ale pewne, trudne do określenia jednym słowem, połączenie: ogólnej strategii rozwiązywania problemów z przekonującym i rzeczywistym przykładem jej użycia. Temu mogą służyć: odpowiednio dobierane przykłady problemów, schematy graficzne algorytmów, symulacje (najlepiej komputerowe) działania rozwiązań, własne obliczenia (mogą być „na piechotę”) itp.

Literatura.

- [1] Kurzewski K.: *Sztuka nauczania. Czynności nauczyciela*. Wydawnictwo Naukowe PWN, 1995r
- [2] Grubiel E., Hardt-Olejniczak G., Krupicka H., Sysło M M.: *Informatyka. Podręcznik dla nauczyciela gimnazjum*. Wydawnictwa Szkolne i Pedagogiczne S.A.,2000r
- [3] Sysło M.M., *Algorytmy*. Wydawnictwa Szkolne i Pedagogiczne 1997r
- [4] Papert S., *Burze mózgów*. Wydawnictwo Naukowe PWN, Warszawa 1996r
- [5] Grubiel E., Hardt-Olejniczak G., Krupicka H., Sysło M M.: *Informatyka. Technologia informacyjna*. Wydawnictwa Szkolne i Pedagogiczne S.A.1999r
- [6] Głowiński C., *Sztuczna inteligencja*. PCkurier Nr 3/1999r
- [7] Sysło M.M., *W labiryncie algorytmów*. Materiał z konferencji pt. Informatyka w szkole, XIV” (Lublin, 16-19.09.1998r)
- [8] Sysło M.M., *Piramidy, szyszki i inne konstrukcje algorytmiczne*. Wydawnictwa Szkolne i Pedagogiczne 1998r
- [9] Grubiel E., Hardt-Olejniczak G., Krupicka H., Sysło M M.: *Edukacja informatyczna w kształceniu ogólnym..* Wydawnictwa Szkolne i Pedagogiczne S.A.1999r