

Podejście zachłanne w rozwiązywaniu problemów. Cz. 1



- **optymalizacja;**
- **algorytm zachłanny;**
- **wydawanie reszty metodą zachłanną;**

Przykładowe problemy optymalizacyjne:

- Warsztat stolarski: jak najmniejsza ilość odpadu produkcyjnego przy jak największej produkcji sztuk mebli;
- Dietetyka: dieta wg proporcji składników odżywczych i oczekiwanej liczby kalorii;

Optymalizacja - metoda wyznaczania najlepszego (optimalnego) rozwiązania z punktu widzenia określonego kryterium (wskaźnika) jakości (np. kosztu, drogi, wydajności).

Rozwiązanie optymalne - najlepsze z możliwych, które umiemy uzasadnić;

Rozwiązanie przybliżone - każde inne od rozwiązywania optymalnego;

Wydawanie reszty metodą zachłanną

Otrzymujesz w sklepie resztę: 8 zł 45 gr wydaną wyłącznie monetami: 1 zł oraz 5 gr czyli dostajesz 17 monet...



Problem wydawania reszty

Dążymy do tego by liczba monet była jak najmniejsza

Metoda wydania najmniejszej liczby monet:

1. Szukamy największego nominału nie większego od reszty;
2. Wydajemy nominał tyle razy ile można by nie przekroczyć reszty;
3. Pomniejszamy resztę do wydania o łączną wartość już wybranych monet, następnie powtarzamy kroki 1 i 2;

Kwota do wydania:

9,45 zł

4,45 zł

0,45 zł

0,05 zł

Nominał:

5 zł

2 zł, 2 zł

20 gr, 20 gr

5 gr



Ćwiczenie 3:

Automat na dworcu kolejowym wydaje resztę najmniejszą możliwą liczbą monet. Ty płacisz banknotem 10 zł za zakup:

- a) wody za 2,70 zł;
- b) herbaty za 3,40 zł;
- c) batona za 4,19 zł;

Z jakich monet składa się reszta w każdym z przykładów?

PROGRAM W C++

Wyznaczający nominały do wydania reszty najmniejszą liczbą monet.

- reszta będzie kwotą niższą niż 10zł;
- Nominały:
 - 5 zł, 2 zł, 1zł, 0.50 zł, 0.20 zł
 - 0.10 zł, 0.05 zł, 0.02 zł, 0.01 zł;

SPECYFIKACJA PROBLEMU:

DANE:

kwota reszty do wydania - liczba z przedziału $[0.01; 9.99]$;
tablica NOMINALY zawierająca uporządkowane malejąco
liczby od 5 do 0.01, odpowiadające dostępnym nominałom

WYNIK:

kolejne wartości nominałów potrzebnych do wydania
reszty;

Lista kroków algorytmu Wydawanie reszty:

1. Wczytaj kwotę reszty do wydania i zapisz ją w zmiennej **reszta**.
2. Ustaw na 0 wartość zmiennej **i** wskazującej komórkę tablicy **NOMINALY**.
3. Dopóki **reszta** > 0, wykonuj kolejno kroki 4-7.
 4. Jeśli **reszta** ≥ **NOMINALY[i]**, to wykonaj kroki 5 i 6.
 5. Wypisz na ekranie wartość **NOMINALY[i]**.
 6. Pomniejsz wartość zmiennej **reszta** o wartość **NOMINALY[i]**.
 7. W przeciwnym wypadku zwiększ **i** o 1.

```
#include <iostream>
using namespace std;
```

```
const int N = 9;
const float NOMINALY[N] = {5, 2, 1, 0.50, 0.20, 0.10, 0.05, 0.02, 0.01};
```

float do wartości zmiennoprzecinkowych

```
void WydajReszta(float reszta)
{
    int i = 0;
    while (reszta>0 && i<N)
    {
        if (reszta >= NOMINALY[i])
        {
            cout << NOMINALY[i] << endl;
            reszta = reszta - NOMINALY[i];
        }
        else i++;
    }
}
```

przed przejściem do kolejnego nominału sprawdzamy warunek while

```
int main()
{
    float reszta;
    cout << "Podaj kwote reszty: ";
    cin >> reszta;
    WydajReszta(reszta);
    return 0;
}
```

Program: WydawanieReszty1 (z float)

ĆWICZENIE:

- Zapisz kod programu jako plik o nazwie:
wydawanie_reszty1_imię_nazwisko_klasa.cpp
- Sprawdź działanie dla wartości reszty: 7.30, 7.79, 7.85.

Liczby w komputerze są przechowywane w postaci binarnej.

np. 0,1 nie ma skończonego rozwinięcia dwójkowego, lecz okresowe: 0,000(1100), dlatego w **zmiennej float** będzie wartość przybliżona, nieco większa niż 0,1.



Błędne wyniki?

- niektóre kwoty reszty są błędne
- np. brakuje 1 grosza (dla 7.85)
- to skutek dodawania wartości float, które są przybliżeniami właściwych wartości
- obliczenia typu zmiennoprzecinkowego obarczone są konsekwencjami **błędów przybliżeń**
- typ **double** też nie załatwi sprawy...
- program mimo niepoprawnego wyniku kończy działanie, nie zawiesza się dzięki **$i < N$** w while

Poprawienie kodu źródłowego dla algorytmu wydawania reszty:

- zastosujemy typ całkowitoliczbowy czyli: **int**
- sprowadzimy wszystkie kwoty reszty ze zł do groszy
- 5 zł = 500
- 2 zł = 200
- 1 zł = 100

POPRAWIONA SPECYFIKACJA PROBLEMU:

DANE:

kwota reszty do wydania w postaci dwóch liczb całkowitych; liczba złotych z przedziału $[0; 9]$ i liczba groszy z przedziału $[0; 99]$; tablica NOMINALY (wielokrotności 1 grosza) uporządkowane malejąco.

WYNIK:

kolejne wartości nominałów potrzebnych do wydawania reszty podane w złotówkach.

```

#include <iostream>
using namespace std;

const int N = 9;
const int NOMINALY[N] = {500, 200, 100, 50, 20, 10, 5, 2, 1};

void WydajReszte(int reszta)
{
    int i = 0;
    while (reszta>0)
    {
        if (reszta >= NOMINALY[i])
        {
            cout << NOMINALY[i]/100.0 << endl;
            reszta = reszta - NOMINALY[i];
        }
        else i++;
    }
}

int main()
{
    int reszta_zlote, reszta_grosze;
    cout << "Podaj liczbe zlotych reszty: ";
    cin >> reszta_zlote;
    cout << "Podaj liczbe groszy reszty: ";
    cin >> reszta_grosze;
    WydajReszte(reszta_zlote*100 + reszta_grosze);
    return 0;
}

```

poprawiony Program: WydawanieReszty2 (z int)

ZADANIE DOMOWE:

- Zapisz kod programu ze slajdu numer: 14 jako plik o nazwie:
wydawanie_reszty2_imię_nazwisko_klasa.cpp
- Sprawdź działanie dla wartości reszty: 7.30, 7.79, 7.85.
- Wyślij metodą wskazaną przez nauczyciela, znaj składnię kodu
- jak zmodyfikować kod by można było do niego wprowadzić kwotę większą niż 10 zł, a resztę wydawać monetami i banknotami?