

Sprawozdanie 2

Zadanie 10. Liczbę naturalną C można przedstawić jako sumę parami różnych liczb naturalnych. Na przykład jeśli $C = 6$, to możemy C przedstawić na cztery sposoby:

$$1 + 2 + 3$$

$$1 + 5$$

$$2 + 4$$

$$6$$

a jeśli $C = 10$, to takimi podziałami są:

$$1 + 2 + 3 + 4$$

$$1 + 2 + 7$$

$$1 + 3 + 6$$

$$1 + 4 + 5$$

$$1 + 9$$

$$2 + 3 + 5$$

$$2 + 8$$

$$3 + 7$$

$$4 + 6$$

$$10$$

Skonstruuj algorytm wyczerpujący z nawrotami, generujący wszystkie podziały podanej liczby naturalnej C .

Adrian Rupala

20 maja 2018

Spis treści

1	Teoria	1
2	Rozwiązanie	1

1. Teoria

Algorytm z nawrotami to algorytm wyszukiwania wszystkich lub kilku rozwiązań. Polega on na znajdowaniu wyniku metodą „prób i błędów”, wszelako z oznaczeniem niepowodzeń, dzięki czemu te same błędy nie są popełniane dwukrotnie.

Jeżeli problem pozwala na zastosowanie algorytmu wyszukiwania z nawrotami, to metoda ta może być znaczenie efektywniejsza niż wyszukiwanie wyczerpujące (zakładające przeszukiwanie wszystkich rozwiązań), ponieważ pojedynczy test może wyeliminować nie jedno, a wiele rozwiązań niedopuszczalnych.

Rekurencja to technika programowania, dzięki której funkcja, procedura lub podprogram jest w stanie w swoim ciele wywołać samą siebie. Pozwala ona łatwo wykonać wiele zadań, w których zachodzi potrzeba obliczenia wyników częściowych do obliczenia całości.

2. Rozwiązanie

Oto algorytm zaimplementowany w języku C++ przedstawiający rozwiązanie problemu przedstawionego w zadaniu:

```
#include <iostream>

using namespace std;

int tablica[100];

void podzial(int C, int obecny, int* tablica, int index) {
    if (obecny + tablica[index] == C) {
        for (int i=0; i <= index; i++) {
            cout << tablica[i] << " ";
        }
        cout << endl;
        ;
    } else if (obecny + tablica[index] > C) {
        ;
    } else {
        for(int i = tablica[index]+1; i < C; i++) {
            tablica[index+1] = i;
            podzial(C, obecny + tablica[index], tablica, index+1);
        }
    }
}

int main(){
    int C = 0;

    cout << "Podaj liczbe: " << endl;
    cin >> C;
    cin.get();
    cout << "=====" << endl;

    for(int i = 1; i <= C; i++) {
        tablica[0] = i;
        podzial(C, 0, tablica, 0);
    }
}
```

```
    cin.get();  
    return 0;  
}
```

Kod programu składa się z dwóch funkcji: `podzial` oraz głównej funkcji `main`.

Funkcja `main` ma za zadanie pobranie od użytkownika liczby, przekazanie jej do zmiennej `C` oraz wypisanie kolejnych tablic, które zawierają pożądaną przez nas liczbę wywołując funkcję `podzial`.

Funkcja `podzial` przyjmuje jako parametry liczbę którą chcemy rozłożyć na czynniki - `C`, obecnie porównywany element - `obecny`, wskaźnik na pierwszy element tablicy - `tablica`, oraz kolejny indeks tablicy - `index`. W pierwszej instrukcji warunkowej `if` następuje sprawdzenie, czy suma obecnego elementu, oraz elementu tablicy jest równa szukanej liczbie, jeśli tak wypisz całą tablicę. Sprawdzenie to ma na celu ustalenie, czy suma elementów rozkładu liczby przechowywana w tablicy osiągnęła liczbę, jaką chcemy rozłożyć. Kolejny warunek sprawdza, czy ta suma jest większa od liczby podanej przez użytkownika, jeśli tak, nie zwraca nic. W każdym innym wypadku za pomocą pętli `for` zwiększamy kolejny element tablicy wtedy. Wtedy gdy jest on mniejszy od podanej przez nas liczby, zwiększamy indeks tablicy o jeden, oraz wywołujemy rekurencyjnie całą funkcję ze zmienionymi parametrami zwiększonego o jeden indeksu, obecnego elementu jako sumy elementu obecnego, oraz aktualnego elementu tablicy.