

# **Zadanie 10. Podział liczby**

Adrian Rupala

---

21 maja 2018

## Treść zadania

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  $C$ .

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 znacznie 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.

```
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;  
}
```

```
Podaj liczbe:  
6  
=====  
1 2 3  
1 5  
2 4  
█
```

**Rysunek 1:** Wynik dla liczby 6.

```
Podaj liczbe:  
10  
=====  
1 2 3 4  
1 2 7  
1 3 6  
1 4 5  
1 9  
2 3 5  
2 8  
3 7  
4 6  
█
```

**Rysunek 2:** Wynik dla liczby 10.

*Dziękuję za uwagę!*