

# Wstęp do programowania\*

II kolokwium

2024/2025

**Uwaga:** każde zadanie proszę rozwiązać na osobnej, podpisanej kartce.

Należy podać złożoność czasową i pamięciową rozwiązań.

1. Dany jest wektor  $n$  dodatnich liczb całkowitych  $v = \{x_0, x_1, \dots, x_{n-1}\}$ . Szukamy takich trójek elementów  $x_i, x_j, x_k$  ( $0 \leq i < j < n$ ,  $0 \leq k < n$ ,  $i \neq k \neq j$ ), że  $x_i + x_j = x_k$ . Napisz funkcję `int trojki(const std::vector<int> v)`, która policzy, ile takich trójek występuje w wektorze  $v$ .

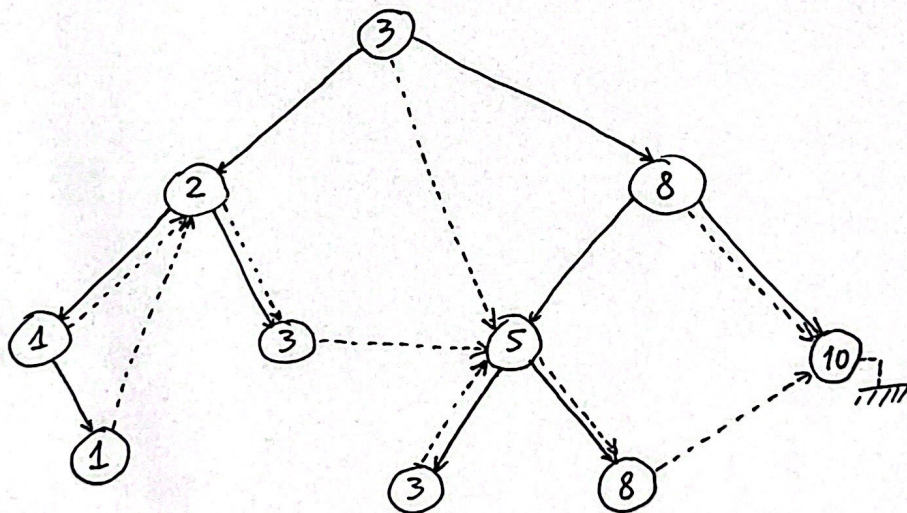
Na przykład, dla danych  $v = \{4, 2, 1, 5, 8, 1\}$ , poprawnym wynikiem jest 3. Mamy jedną trójkę postaci  $1 + 1 = 2$  i dwie trójki postaci  $1 + 4 = 5$ .

2. Dana jest definicja drzew binarnych, z dodatkowym wskaźnikiem (`next`) w każdym węźle:

```
typedef struct node *bin_tree;
struct node {
    int val;
    bin_tree left, right, next;
};
```

Mamy dane drzewo BST, w którym wartości w węzłach mogą się powtarzać, a wskaźniki `next` mają dowolne wartości (nie są zainicjowane).

Napisz funkcję `void fastryguj(bin_tree t)`, która poustawia w węzłach wskaźniki `next` tak, żeby każdy z nich wskazywał na najbliższy (w sensie in-order) węzeł zawierający ściśle większą wartość. Jeśli taki węzeł nie istnieje, to wskaźnik `next` należy ustawić na `NULL`.



Na rysunku, linią przerywaną zaznaczono jak należy ustawić wskaźniki `next`.