

# Algorytmy i struktury danych

## Lista 7

### Zadanie 1.

Pokaż w jaki sposób można efektywnie przetrzymywać kopiec  $d$ -arny rozmiaru  $n$  w tablicy długości  $n$ .

### Zadanie 2.

Pokaż w jaki sposób zbudować kopiec  $d$ -arny z losowej tablicy długości  $n$  w czasie liniowym od wielkości danych.

### Zadanie 3.

Pokaż w jaki sposób można zaimplementować operacje kolejki priorytetowej: Insert, Minimum, ExtractMin, DecreaseKey, Union, Delete przy użyciu struktury drzewa BST. Zbadaj złożoność obliczeniową tych operacji oraz porównaj je ze złożonościami implementacji operacji kolejki priorytetowej na kopcu binarnym.

### Zadanie 4.

Mamy dany spójny graf  $G$  oraz wyróżniony wierzchołek  $v$ . Wykonujemy procedury DFS i BFS, zaczynając w  $v$  i okazuje się, że obie tworzą to samo drzewo przejścia  $T$ . Pokaż, że  $G = T$ .

### Zadanie 5.

Czy prawdziwe jest następujące stwierdzenie? Pokaż kontrprzykład lub udowodnij.

Niech  $G$  będzie grafem o  $n$  wierzchołkach, gdzie  $n$  jest parzyste. Jeśli każdy wierzchołek ma stopień przynajmniej  $n/2$ , to graf jest spójny.

### Zadanie 6.

Jaki będzie czas działania procedury BFS, jeśli graf wejściowy jest reprezentowany przez macierz sąsiedztwa, a algorytm jest zmodyfikowany w taki sposób, żeby działał poprawnie dla tej reprezentacji?

### Zadanie 7.

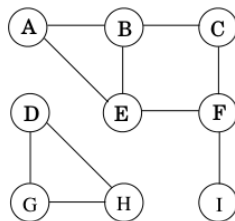
Dla pewnego grafu skierowanego  $G = (V, E)$ , niech  $G^R = (V, E^R)$  będzie takim grafem, że

$$(u, v) \in E^R \text{ wtedy i tylko wtedy gdy } (v, u) \in E$$

(czyli grafem o tych samych wierzchołkach, ale z odwróconymi kierunkami wszystkich krawędzi). Pokaż jak w złożoności liniowej od wielkości grafu wyznaczyć reprezentację list sąsiedztwa grafu  $G^R$  mając daną reprezentację list sąsiedztwa grafu  $G$ .

### Zadanie 8.

Zaprezentuj działanie procedury DFS na następującym grafie. Podaj wartości `pre` i `post` dla każdego wierzchołka. Zaprezentuj również działanie procedury BFS zaczynając od wierzchołka  $A$ .



### Zadanie 9.

Zaprezentuj algorytm wyszukiwania węzła z zadaniem kluczem w kopcu binarnym (ale nie polegający na przeszukaniu całej tablicy zawierającej kopiec). Wykonaj analizę typu worst case zaprezentowanego algorytmu oraz wykonaj symulację komputerową średniej złożoności swojego algorytmu oraz algorytmu przeszukującego w standardowy sposób całą tablicę zawierającą kopiec (symulacja powinna polegać na wywołaniu obu algorytmów dla losowego kopca o rozmiarze  $n$  dla  $n = 10, \dots, 1000$  z krokiem  $step = 11$ , dla każdego  $n$  po 100 razy).