

ALGORYTMY I STRUKTURY DANYCH

ZŁĄCZALNE KOLEJKI PRIORYTETOWE

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

1. [*] Do początkowo pustego kopca dwumianowego wstawiamy metodą `insert()` następujące liczby: 23, 11, 37, 41, 19, 17 i 43. Potem dwukrotnie wyciągamy element maksymalny metodą `extract-max()`. Pokaż, jak będzie się zmieniać kopiec po kolejnych operacjach.
2. [**] Jak włożyć drzewo dwumianowe k -tego stopnia w tablicę 2^k elementową? Napisz wzory pozwalające określić indeks przodka i indeksy następników węzła o zadanym numerze. Dlaczego nie stosuje się reprezentacji tablicowej drzew dwumianowych w kopcach dwumianowych?
3. [**] Załóżmy, że lista korzeni drzew dwumianowych w kopcu dwumianowym jest uporządkowana malejąco według stopni (zamiast rosnąco). Pokaż, jak w takim przypadku zaimplementować operacje kopca złączalnego, aby nie pogorszyć asymptotycznego czasu działania poszczególnych procedur.
4. [***] Dany jest spójny graf nieskierowany $G(V, E)$ oraz funkcja wag krawędzi $w : E \mapsto \mathbf{R}$. Należy znaleźć minimalne drzewo rozpinające w grafie G , czyli acykliczny podzbiór $T \subseteq E$, który łączy wszystkie wierzchołki z V i którego całkowita waga $w(T) = \sum_{e \in T} w(e)$ jest najmniejsza. W twoim algorytmie zastosuj *kopce złączalne*!
5. [***] Dowolne drzewo ukorzenione, w tym również drzewo dwumianowe, jest można implementować za pomocą węzłów, w których pamiętane są trzy wskaźniki: *left-child* (najbardziej lewy syn), *right-sibling* (prawy brat) i *parent* (ojciec). Taka reprezentacja drzewa nosi nazwę “*na lewo syn, na prawo brat*”. Dzięki temu z każdego węzła jest dostęp do ojca. Wskaż, jak zachować tę własność, pamiętając w każdym węźle tylko dwa wskaźniki oraz jedną wartość boolowską.
6. [***] Napisz nierekurencyjną procedurę działającą w liniowym czasie $O(n)$, która danego drzewa binarnego o n węzłach wypisze wszystkie jego klucze w porządku *in-order*. Twoja procedura może wykorzystać tylko stałą liczbę $O(1)$ dodatkowych komórek pamięci. Pierwotna struktura drzewa powinna zostać niezmieniona.
7. [*] Napisz nierekurencyjną procedurę działającą w liniowym czasie $O(n)$, która odwraca kolejność elementów na jednokierunkowej liście o n węzłach. Twoja procedura może wykorzystać tylko stałą liczbę $O(1)$ dodatkowych komórek pamięci.

8. [***] Jak zaimplementować listę dwukierunkową, używając tylko jednego wskaźnika np zamiast dwóch $next$ i $prev$?

Uwaga: Należy założyć, że wszystkie wskaźniki są reprezentowane za pomocą k -bitowych nieujemnych liczb całkowitych.

Wskazówka: Zdefiniuj np jako $next \oplus prev$ (operacja XOR). Wskaźnik pusty jest reprezentowany przez liczbę 0.