ćwiczenia (licencjat dzienny): lista zadań nr 9 *11 grudnia 2006 r.*

ALGORYTMY I STRUKTURY DANYCH

KOLEJKI PRIORYTETOWE

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

1. [∗∗] Na wykładzie zdefiniowany został kopiec binarny oraz przedstawiona została jego implementacja tablicowa. Uogólnij pojęcie kopca na kopce d-arne, dla dowolnego $d \geq 2$. Jak taki kopiec włożyć w tablicę i jak potem wyliczać indeksy synów i ojca dla danego elementu? Zilustruj działanie twoich procedur na przykładzie kopca 3-arnego lub 4-arnego.
   Wskazówka: Użyj tablicy indeksowanej od 0.

2. [∗∗] *Kopiec minimaksowy* to struktura danych podobna do kopca, której zadaniem jest efektywna realizacja operacji kolejkowych *insert* i *extract-max* oraz dodatkowo *extract-min*. Opracuj implementację kopca realizującą wszystkie wymienione operacje w czasie logarytmicznym względem liczby elementów.
   Wskazówka: Trzymaj dwa kopce — jeden z maksimum w korzeniu a drugi z minimum. Jak umieścić je w jednej tablicy wypełnionej spójnie od początku?

3. [∗] Wykaż, że w drzewie dwumianowym stopnia k, które zawiera 2^k węzłów, długość ścieżki od korzenia do dowolnego węzła w tym drzewie jest nie większa niż k.

4. [∗∗] Jak zaimplementować drzewo dwumianowe k-tego stopnia w tablicy 2^k elementowej? Twoje rozwiązanie powinno pozwalać na łatwe łączenie dwóch drzew dwumianowych. Napisz wzory pozwalające ustalić na podstawie indeksu w tablicy numer poziomu, na którym węzeł się znajduje, jego stopień oraz indeksy ojca, brata i kolejnych synów.

5. [∗∗∗] Dane jest drzewo dwumianowe stopnia k, które zawiera 2^k węzłów. W drzewie tym jest zachowany porządek kopcowy. Następnie zmieniamy w dowolnym wybranym węźle pamiętaną w nim wartość. Porządek kopcowy może więc zostać zaburzony. Jeśli zmienimy wartość na większą, to zwykłe *sianie w górę* przywróci ten porządek w $O(\log n)$ krokach, a więc w akceptowalnym czasie. Jeśli natomiast zmienimy wartość na mniejszą, to *sianie w dół* przywróci porządek kopcowy w $O(\log^2 n)$ krokach, a to jest zdecydowanie za długo. Opracuj metodę, która po zmianie wartości na mniejszą we wskazanym elemencie przywróci porządek kopcowy w czasie $O(\log n)$.

6. [∗] Wykaż, że dla asymptotycznego czasu działania wszystkich operacji na kopcu dwumianowym nie ma znaczenia, czy lista korzeni drzew dwumianowych jest uporządkowana rosnąco czy malejąco.
   Wskazówka: Jak zaimplementować procedurę działającą w liniowym czasie $O(n)$ w miejscu, która odwraca kolejność elementów na jednokierunkowej liście o n węzłach?

7. [**] Jak zaimplementować operację *insert* na kopcu dwumianowym, aby wstawianie elementu do kopca odbywało się bezpośrednio (bez konieczności użycia procedury *union*).
Wskazówka: Przeanalizuj związek między wstawianiem elementu do kopca dwumianowego a zwiększaniem o 1 liczby binarnej.

8. [***] *Drzewem lewicowym* nazywamy drzewo binarne, w którym dla każdego wierzchołka x prawa skrajna ścieżka od x do liścia jest najkrótszą ścieżką od x do wskaźnika pustego w poddrzewie zakorzenionym w x. Opracuj implementację złączalnej kolejki priorytetowej, korzystającą z drzew lewicowych. Złożoność operacji kolejkowych *insert*, *extract-max* i *union* powinna być logarytmiczna względem liczby wszystkich elementów w kolejce.
Wskazówka: Zastanów się, jak połączyć dwa kopce lewicowe wzdłuż prawych skrajnych ścieżek.