ćwiczenia (licencjat wieczorowy): lista zadań nr 9 *19 maja 2009 r.*

ALGORYTMY I STRUKTURY DANYCH

KOLEJKI PRIORYTETOWE

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

1. [1 pkt.] Pokaż, jak za pomocą kolejki priorytetowej zaimplementować zwykłą kolejkę *FIFO* i stos *LIFO*.

2. [1 pkt.] Wykaż, że odwrotnie posortowana tablica jest kopcem.

3. [1 pkt.] Wykaż, że sortowanie kopcowe *heap–sort* nie jest stabilne.

4. [2 pkt.] 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– lub 4–arnego.

   Wskazówka. Użyj tablicy indeksowanej od 0.

5. [3 pkt.] *Kopiec minimaksowy* to struktura danych podobna do kopca, której zadaniem jest efektywna realizacja operacji kolejkowych *insert*, *extract–max* oraz dodatkowo *extract–min*. Opracuj implementację kopca realizującą wszystkie wymienione operacje w czasie logarytmicznym względem liczby elementów. Jak zbudować taki kopiec w czasie liniowym?

   Wskazówka. Trzymaj dwa równoliczne z dokładnością do 1 elementu kopce — jeden z maksimum a drugi z minimum w korzeniu (elementy w obu kopcach nie dublują się). Jak umieścić je w jednej tablicy wypełnionej spójnie od początku?

6. [1 pkt.] 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.

7. [2 pkt.] 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.

8. [3 pkt.] 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 przywrócenie porządku kopcowwego poprzez proste *sianie w dół* będzie działać istotnie dłużej ($O(\log^2 n)$ kroków). Opracuj metodę, która po zmianie wartości na mniejszą we wskazanym elemencie przywróci porządek kopcowy w czasie $O(\log n)$.