Algoryty i Struktury Danych 2013

Lista 4

1. Udowodnij, że każda operacja na kopcu złączalnym (dwumianowym), zawierającym n kluczy, ma złożoność $O(\log n)$.

2. Narysuj przykładowy kopiec dwumianowy o 17 kluczach. Wykonaj 4 razy operację `getmax`.

3. Jak jest minimalna a jaka maksymalna liczba kluczy w drzewie czerwono-czarnym o czarnej wysokości h_b.

4. Do pustego drzewa czerwono-czarnego wstaw kolejno litery swojego imienia i nazwiska. Następnie usuń je w tej samej kolejności, w jakiej były wstawiane.

5. Oznaczmy przez $f(n)$ ilość kształtów drzew BST o n węzłach. Napisz wzór wyrażający $f(n)$ przez $f(0), \ldots, f(n-1)$. Napisz funkcję, która wypełnia tablicę `int f[n]` wartościami tej funkcji. Jaka jest złożoność twojej procedury. Jaka byłaby złożoność w przypadku zastosowania rekurencji?

6. Jaką dodatkową informację należy przechowywać w każdym węźle drzewa binarnego, by łatwo znajdować elementy według rangi? Napisz funkcję `BTSnode* ity(BSTnode *t, int i)`, korzystającą z tego dodatkowego pola, która będzie działała w czasie $O(\log n)$ dla drzew zrównoważonych.

7. Jaką informację trzeba dodać do każdego węzła, by w czasie $O(\log n)$ losować z drzewa klucze z prawdopodobnieństwem proporcjonalnym do klucza.

8. Napisz funkcję rekurencyjną `int F(BSTnode * t)`obliczającą:

   (a) ilość węzłów w drzewie `t`,

   (b) głębokość drzewa `t`,

9. Napisz implementację usuwania węzła z drzewa binarnego wg następującego schematu:

   (a) jeśli usuwany węzeł nie ma dzieci, to go usuwamy a odpowiedni wskaźnik zmieniamy na NULL.

   (b) jeśli ma jedno dziecko, to go usuwamy, a odpowiedni wskaźnik w węźle rodzica zastępujemy wskaźnikiem na to dziecko.

   (c) jeśli ma dwoje dzieci, to nie usuwamy tego węzła, lecz najmniejszy element w jego prawym poddrzewie, a dane i klucz tego elementu wpisujemy do węzła, który miał być usunięty.