

Algorytmy i Struktury Danych - lista 4

1. Napisz funkcję obliczającą x^n w czasie $O(\log n)$ w wersjach z użyciem i bez użycia rekurencji.
2. Jaka byłaby złożoność algorytmu `merge_sort` gdyby dzielić tablice nie na równe części, lecz na części o wielkości 25% i 75%, lub ogólniej, na części o rozmiarach nq oraz $n(1 - q)$.
3. Napisz algorytm `merge_sort`, który działałby na liście 1-linkowanej.
4. Oznaczmy przez $f(n)$ ilość kształtów drzew BST o n węzłach. Napisz wzór wyrażający $f(n)$ przez $f(0), \dots, f(n - 1)$.
5. Napisz funkcję rekurencyjną `int F(BSTnode * t)` obliczającą:
 - (a) ilość węzłów w drzewie `t`,
 - (b) głębokość w drzewa `t`,
6. 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.
7. Jaką dodatkową informację należy przechowywać w każdym węźle drzewa binarnego, by łatwo znajdować medianę zawartych w nim elementów? Napisz implementację funkcji `BSTnode* ity(BSTnode *t, int i)`, korzystając z tego dodatkowego pola, która będzie działała w czasie $O(\log n)$ dla drzew zrównoważonych.
8. Jak wyliczyć współczynniki c_0, c_1, c_2 wielomianu $c_2x^2 + c_1x + c_0 = (a_1x + a_0)(b_1x + b_0)$ znając a_0, a_1, b_0, b_1 używając tylko trzech mnożeń.
9. Wielomian stopnia $2n$ można pomnożyć używając 4 mnożeń wielomianów stopnia n . Jaka będzie asymptotyczna złożoność takiego algorytmu? Jak zmieni się ona gdy zamiast 4 użyjemy 3 mnożeń, wykorzystując pomysł z poprzedniego zadania.