

## Algorytmy - Lista 2

1. Napisz funkcję rekurencyjną `int F(BSTnode * t)` obliczającą:
  - (a) ilość węzłów w drzewie `t`,
  - (b) głębokość w drzewa `t`,
2. Oznaczmy przez  $f(n)$  ilość kształtów drzew BST o  $n$  węzłach. Napisz wzór wyrażający  $f(n)$  przez  $f(0), \dots, f(n-1)$ .
3. 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 węzle 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.
4. Jak jest minimalna ilość porównań gwarantująca posortowanie  $n$  liczb dla  $n = 2, n = 3, n = 4$  oraz  $n = 5$ . (Nadobowiązkowe  $n = 6, n = 7$ ).
5. Jak wiąże się faktyczna ilość porównań wykonywanych przez algorytm sortowania przez proste wstawianie z ilością inwersji w danych wejściowych?
6. Znajdź związek między ilością porównań faktycznie wykonywanych przy sortowaniu przez wstawianie tablicy  $a_0, \dots, a_{n-1}$  a ilością inwersji w tej tablicy. (Inwersją nazywamy parę indeksów  $(i, j)$  taką, że  $i < j \wedge a_i > a_j$ ).
7. Użyj rekursji do rozwiązania problemu wież z Hanoi.
8. Napisz program rozwiązujący problem wież z Hanoi: Jaka jest liczba ruchów potrzebna do przełożenia  $n$  krążków różnej wielkości z patyczka A na patyczek C, jeśli można używać też patyczka B, a krążek większy nie może nigdy leżeć na mniejszym. Program powinien wypisywać wszystkie konieczne ruchy w postaci:
  1. A->C
  2. A->B
  3. C->B

9. Napisz program znajdujący wszystkie ustawienia 8 hetmanów na szachownicy takie, że żaden z nich nie szachuje innego.
10. Niech dana będzie tablica A zawierająca pary liczb:  $(1, 0)$ ,  $(2, 100)$ ,  $(3, 100) \dots (n, 100)$ . Pierwszy element każdej pary nazywamy nazwą, a drugi wartością. Tablicę tą traktujemy jako kopiec ze względu na wartość. Jaką dodatkową strukturę danych trzeba użyć, by w efektywny sposób wykonywać operację `Decrease_value(k, x)`, przypisującą nazwie  $k$  nową wartość  $x$  mniejszą od poprzedniej? Jak należy zmodyfikować operacje `Delete_min` oraz `Insert`, by ta nowa struktura nie dezaktualizowała się w trakcie ich działania?
11. Jak skonstruować program liczący ułamek łańcuchowy

$$g = \frac{a}{b + \frac{a}{b + \frac{a}{b + \dots}}}$$

z zadaną z góry dokładnością.

12. Udowodnij, że  $Z_7$  jest ciałem tzn. w pierścieniu  $\{0, 1, \dots, 6\}$  z mnożeniem i dodawaniem modulo 7 jest jednoznacznie zdefiniowana operacja odwrotna do mnożenia. Uzasadnij, że jest tak dla każdego  $Z_p$  gdy  $p$  jest liczbą pierwszą.
13. Zakładając, że masz do dyspozycji algorytm na mnożenie dowolnie długich liczb całkowitych oraz znasz wartość pewnej dużej liczby pierwszej (np. 100-cyfrowej) zaproponuj sposób doskonałego szyfrowania tekstu o długości do 100 znaków i niedoskonałego szyfrowania tekstów dłuższych.