

Algorytmy i Struktury Danych

Egzamin poprawkowy - 10 września 2021 (część I)

1. (2pkt) Dana jest definicja struktury węzła listy pojedynczo związanej:

```
struct lnode {  int key;  lnode* next; };
```

- (a) Napisz funkcję `void remove_greater(lnode*& L, int x)` usuwającą z listy `L` wszystkie elementy większe od `x`.
- (b) Napisz funkcję `void append(lnode* &L, int x)`, która dodaje liczbę `x` na końcu listy `L`.

2. (3pkt) Dana jest definicja struktury węzła drzewa BST:

```
struct node{  int x;    node* left;    node* right; };
```

- (a) Napisz funkcję `int max(node* t)` zwracającą największy klucz drzewa BST o korzeniu `t`. Jeśli drzewo jest puste, to wynikiem powinno być 0.
- (b) Napisz funkcję `void insert(node*& t, int x)` wstawiającą do drzewa BST o korzeniu `t` nowy klucz `x`.

3. (3 pkt) Zastosuj twierdzenie o rekursji uniwersalnej do rozwiązania zależności:

(a) $T(n) = T(n/3) + \sqrt{n}$

(b) $T(n) = 4T(n/2) + n^2$

(c) $T(n) = 6T(n/3) + n^2$

(d) $T(n) = 9T(n/2) + n^3$

4. Algorytm **HeapSort**.

- (a) (1pkt) Podaj wzory na indeksy lewego syna, prawego syna oraz ojca elementu `t[i]`.
- (b) (1pkt) Napisz funkcję `void przesiej(int t[], int i, int n)` i podaj jej złożoność czasową i pamięciową z uzasadnieniem.
- (c) (3pkt) Załóż, że w tablicy znajdują się 4 ostatnie cyfry numeru albumu i 5 ostatnich cyfr Twojego peselu. Narysuj tę tablicę i początkowy stan kopca. Następnie zasymuluj działanie algorytmu **HeapSort** rysując stan kopca po każdym wywołaniu procedury `przesiej`. Na koniec narysuj wynikowy stan tablicy.
- (d) (1pkt) Czy algorytm jest stabilny? Dlaczego?

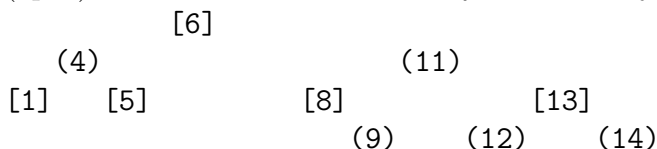
5. (4pkt) Sortowanie szybkie:

- (a) Napisz funkcję `int partition(double t[], int n)`;
- (b) Co jest wynikiem funkcji `partition`? Jaka jest jej złożoność?
- (c) Napisz funkcję `void quick_sort(double t[], int n)` (sortowanie szybkie).
- (d) Jaka jest złożoność `quick_sort` (średnia i pesymistyczna)? Jaka jest maksymalna głębokość rekursji? Odpowiedź uzasadnij.

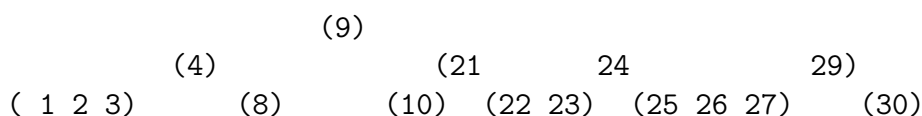
Algorytmy i Struktury Danych

Egzamin poprawkowy - 10 września 2021 (część II)

6. (2pkt) Dla tablicy z haszowaniem podwójnym o rozmiarze $m = 19$ i funkcjach haszujących $h_1(x) = (x \bmod 17) + (x \bmod 3)$ oraz $h_2(x) = x \bmod 18 + 1$ wyznacz ciąg kontrolny dla dwucyfrowej końcówki Twojego albumu.
7. (a) (1pkt) Wymień i opisz jakie informacje zapisane są w węźle kopca dwumianowego.
(b) (2pkt) Jaki kopiec dwumianowy otrzymasz wstawiając do pustego kopca kolejno litery swojego nazwiska, imienia i miasta zamieszkania (poprzestań na pierwszych 13 literach). Przyjmij wariant kopca, gdzie na szczycie są litery z początku alfabetu.
8. (3pkt) W drzewie czerwono-czarnym czarne węzły oznaczono nawiasem kwadratowym:



- (a) wstaw do powyższego drzewa 10.
(b) wstaw do powyższego drzewa 15.
(c) usuń z powyższego drzewa 5.
9. (3pkt) Na rysunku poniżej przedstawiono B-drzewo o $t = 2$.



- (a) Usuń z niego klucz 8.
(b) Z drzewa jakie otrzymałeś usuń 10.
(c) Do drzewa widocznego w treści zadania wstaw klucz 31.
10. Dla następującej listy krawędzi grafu nieskierowanego:
(0,1):8, (0,5):13, (1,2):11, (1,4):2+ c_1 , (2,3):30, (3,4):8+ c_2 , (3,8):5, (4,5):1+ c_3 , (4,7):15,
(5,6):1+ c_4 , (6,7):3+ c_5 , (6,9):11, (7,8):2, (8,9):4+ c_6 ,
gdzie $c_1, c_2, c_3, c_4, c_5, c_6$, to cyfry Twojego albumu, wykonaj co następuje:

- (a) (1pkt) Napisz numer albumu, wypisz obliczone wartości wag krawędzi i wykonaj rysunek grafu. Zapisz tablicę list sąsiedztwa. Wierzchołki na listach sąsiedztwa powinny być są ustawione rosnąco wg numeru wierzchołka. Ta kolejność powinna być stosowana w symulacji algorytmów BFS i Prima.
- (b) (1pkt) Zapisz kolejność odwiedzania wierzchołków w algorytmie BFS startującym z wierzchołka o numerze równym **przedostatniej** cyfrze Twojego albumu.
- (c) (2 pkt) Zasymuluj działanie algorytmu Prima startując z wierzchołka o numerze równym **ostatniej** cyfrze Twojego albumu) i zilustruj rysunkiem:
- strzałkami ciągłymi oznacz krawędzie należące do drzewa wynikowego,
 - liniami przerywanymi oznacz pozostałe krawędzie grafu,
 - przy każdym wierzchołku w nawiasie okrągłym podaj kolejność w jakiej był odwiedzony.