

# Algorytmy i Struktury Danych

## Egzamin - 28 czerwca 2021 (część I)

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

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

- (a) Napisz funkcję `int get_first(lnode* &L)`, która usuwa pierwszy element niepustej listy `L` i zwraca wartość usuniętego klucza.
- (b) Napisz funkcję `int increment_x(lnode* L, int x)`, która na liście `L` wszystkie elementy równe `x` zwiększa o 1. Wynikiem zwracanym przez funkcję, jest ilość zwiększonych elementów.

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

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

- (a) Napisz funkcję `void remove_min(node*& T)`, która usunie z niepustego drzewa `T` element minimalny. Funkcja może być rekurencyjna lub nierekurencyjna.
- (b) Napisz rekurencyjną funkcję `int even_minus_odd(node* T)`, której wynikiem jest ilość parzystych kluczy w drzewie `T` pomniejszona o ilość kluczy nieparzystych.

3. (2pkt) Jaka jest średnia a jaka pesymistyczna złożoność wyszukania klucza najlepszym znanym Ci algorytmem w  $n$ -elementowej: (a) tablicy posortowanej (b) tablicy z haszowaniem (c) drzewie BST (d) drzewie czerwono czarnym ?

4. (3 pkt) Zastosuj twierdzenie o rekursji uniwersalnej do rozwiązania zależności:

- (a)  $T(n) = T(n/2) + \sqrt{n}$
- (b)  $T(n) = 3T(n/2) + n^2$
- (c)  $T(n) = 5T(n/2) + n^2$
- (d)  $T(n) = 8T(n/4) + n\sqrt{n}$
- (e)  $T(n) = 3T(n/3) + n$
- (f)  $T(n) = 10T(n/3) + n^2$

5. (4pkt) Mergesort:

- (a) Napisz funkcję:  

```
void merge(double t1[], int n1, double t2[], int n2, double t3[]),
```

która łączy posortowane tablice `t1`, `t2` o rozmiarach `n1`, `n2` w jedną posortowaną tablicę `t3`.
- (b) Ile maksymalnie a ile minimalnie porównań między elementami tablicy może wykonać funkcja `merge`, jeśli `n1=130`, a `n2=150`? Podaj dokładne wartości.
- (c) Napisz w jednym zadaniu, na czym polega rekurencyjny algorytm `merge_sort`.
- (d) Jaka jest złożoność czasowa `merge_sort`? Udowodnij swoją odpowiedź korzystając z twierdzenia o rekursji uniwersalnej.

6. (4pkt) Sortowanie w czasie liniowym:

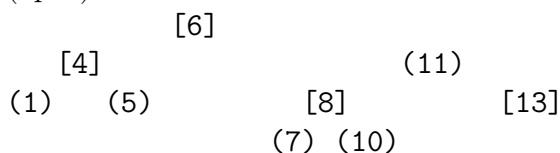
- (a) Napisz funkcję `void counting_sort(double t[], int n)`.
- (b) Jaka jest złożoność pamięciowa  $M(n)$  a jaka czasowa  $T(n)$  funkcji `counting_sort`, jeśli wiadomo, że elementy tablicy są liczbami naturalnymi mniejszymi od  $m$ .
- (c) W jednym zdaniu można ująć działanie procedury `radix_sort`. Napisz to zdanie.
- (d) Jaka jest pesymistyczna złożoność czasowa `radix_sort` dla  $n$  elementowej tablicy liczb  $k$ -cyfrowych?

# Algorytmy i Struktury Danych

## Egzamin - 28 czerwca 2021 (część II)

7. (2pkt) Dla tablicy z haszowaniem podwójnym o rozmiarze  $m = 13$  i funkcjach haszujących  $h_1(x) = (x \bmod 11) + (x \bmod 3)$  oraz  $h_2(x) = x \bmod 12 + 1$  wyznacz ciąg kontrolny dla dwucyfrowej końcówki Twojego albumu.
8. (2pkt) Ile bitów zaoszczędzisz korzystając z kodów Huffmana zamiast kodów o stałej długości, jeśli w kodowanym tekście są tylko litery: a, b, c, d, e. Ich ilości wystąpień otrzymasz dodając jedynki do pięciu ostatnich cyfr swojego albumu.
9. (2pkt) Ilość elementów w kopcu dwumianowym pokrywa się z 3-cyfrową końcówką Twojego numeru albumu. Z ilu drzew dwumianowych składa się ten koniec dwumianowy? Wypisz rzędy tych drzew w kolejności zgodnej z kolejnością w kopcu.
10. (2pkt) Narysuj kopiec dwumianowy zawierający kwadraty cyfr Twojego albumu oraz liczbę 2, tak aby widoczne były wszystkie wskaźniki (jako strzałki) oraz wartości pól fizycznie występujące w węzłach kopca.

11. (2pkt) Drzewa czerwono czarne:

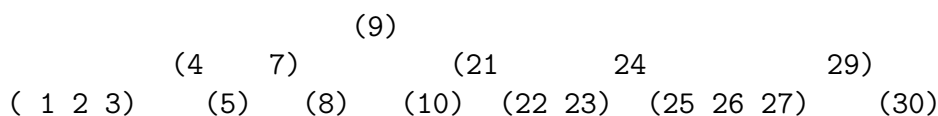


(a) wstaw do powyższego drzewa 9.

(b) usuń z powyższego drzewa 13.

12. (2pkt) Narysuj poprawne drzewo czerwono-czarne o wysokości  $h = 4$ , które na lewo od korzenia ma tylko jeden węzeł. O ile maksymalnie elementów może się różnić ilość elementów w lewym i prawym podrzewie drzewa o czarnej wysokości  $h_B$ ?

13. (3pkt) Na rysunku poniżej przedstawiono B-drzewo o  $t = 2$ .



(a) Usuń z niego klucz 8.

(b) Z drzewa jakie otrzymałeś usuń 5.

(c) Do drzewa widocznego w treści zadania wstaw klucz 28.

14. (1pkt) Oblicz z definicji dyskretną transformatę Fouriera ciągu złożonego z ostatnich 4 cyfr Twojego albumu.