

## GRUPA A

**[10 pkt.] Zadanie 1.** Dana jest struktura danych

```
struct Point { double x,y; };
```

Opisująca punkty w przestrzeni  $R^2$ . Proszę zaimplementować funkcję

```
void heapsort(Point* A, int n);
```

Która otrzymuje na wejściu  $n$  elementową tablicę struktur typu Point i sortuje ją w kolejności rosnącej względem odległości punktu od początku układu współrzędnych, korzystając z algorytmu heapsort.

**[10 pkt.] Zadanie 2.** Proszę zaimplementować algorytm, który mając na wejściu dwa drzewa BST (przechowujące liczby typu int; proszę zadeklarować odpowiednie struktury danych) zwraca nowe drzewo BST, zawierające wyłącznie te liczby, które występują w obu drzewach. Algorytm powinien być jak najszybszy i wykorzystywać jak najmniej pamięci. Jaka jest złożoność zaproponowanego algorytmu? Co można powiedzieć o zrównoważeniu drzew tworzonych przez zaproponowany algorytm?

**[10 pkt.] Zadanie 3.** Niech  $G = (V, E)$  będzie pewnym spójnym nieskierowanym grafem. Dla każdych dwóch wierzchołków  $u, v \in V$ , przez  $d(u, v)$  rozumiemy długość najkrótszej ścieżki między  $u$  i  $v$  (mierzoną liczbą krawędzi). Długością przekątnej grafu  $G$  nazywamy wartość  $\max_{u,v \in V} d(u, v)$ . Proszę opisać możliwie jak najszybszy algorytm, który mając na wejściu acykliczny graf nieskierowany (reprezentowany przez listy sąsiedztwa) oblicza długość jego przekątnej (podpowiedź: nasz graf jest dość szczególnej postaci, co bardzo ułatwia zadanie).

## GRUPA B

**[10 pkt.] Zadanie 1.** Dana jest struktura danych

```
struct Rectangle { double x,y; double w,h; };
```

Opisująca prostokąty (pola  $x$  i  $y$  to współrzędne lewego górnego rogu prostokąta a  $w$  i  $h$  to jego wysokość i szerokość). Proszę zaimplementować funkcję

```
void heapsort(Rectangle* A, int n);
```

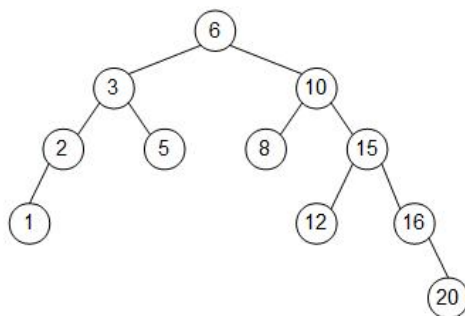
która otrzymuje na wejściu  $n$  elementową tablicę struktur typu Rectangle i sortuje ją w kolejności rosnącej względem wartości pola prostokąta, korzystając z algorytmu heapsort.

**[10 pkt.] Zadanie 2.** Proszę zaimplementować algorytm, który mając na wejściu dwa drzewa BST (przechowujące liczby typu int; proszę zadeklarować odpowiednie struktury danych) zwraca nowe drzewo BST zawierające wyłącznie te liczby, które występują w dokładnie jednym z drzew (ale nie w obu). Algorytm powinien być jak najszybszy i wykorzystywać jak najmniej pamięci. Jaka jest złożoność czasowa zaproponowanego algorytmu? Co można powiedzieć o zrównoważeniu drzew tworzonych przez zaproponowany algorytm?

**[10 pkt.] Zadanie 3.** Niech  $G = (V, E)$  będzie pewnym spójnym nieskierowanym grafem. Dla każdych dwóch wierzchołków  $u, v \in V$ , przez  $d(u, v)$  rozumiemy długość najkrótszej ścieżki między  $u$  i  $v$  (mierzoną liczbą krawędzi). Długością przekątnej grafu  $G$  nazywamy wartość  $\max_{u,v \in V} d(u, v)$ . Proszę opisać możliwie jak najszybszy algorytm, który mając na wejściu acykliczny graf nieskierowany (reprezentowany przez listy sąsiedztwa) oblicza długość jego przekątnej (podpowiedź: nasz graf jest dość szczególnej postaci, co bardzo ułatwia zadanie).

## GRUPA A

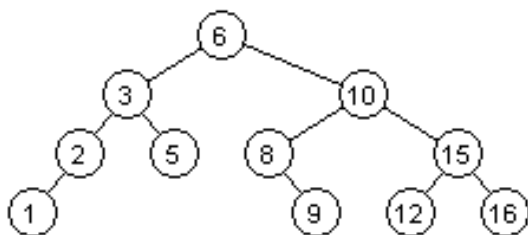
1. Kiedy algorytm o złożoności  $O(n \log n)$  może być lepszy od algorytmu o złożoności  $O(n)$ ?
2. W jakich sytuacjach drzewa czerwono-czarne są lepsze od tablic haszujących?
3. Zaimplementować funkcję, która znajduje i usuwa najmniejszy element z listy dwukierunkowej. Zapisz potrzebne struktury.
4. Podaj reguły kolorowania drzew czerwono-czarnych. Pokoloruj drzewo:



5. Rozważamy zagadnienie najkrótszych ścieżek w grafie i algorytmu BFS, Dijkstry i Bellmana-Forda. Podaj w jakich sytuacjach używamy każdego z wymienionych.
6. Jakich struktur danych użyć dla reprezentacji zbiorów rozłącznych, tak aby operacje FindSet i operacja wykonywania sumy mnogościowej były jak najszybsze.
7. Kiedy reprezentacja grafu przez listy sąsiedztwa jest lepsza od reprezentacji na macierzy sąsiedztwa?
8. Zaimplementuj BFS dla grafu w postaci macierzowej.
9. Podaj przykład algorytmu zachłannego i algorytmu dynamicznego. Krótko opisz zasadę działania jednego z nich.
10. Podaj ideę SkipListy.

## GRUPA B

1. Kiedy algorytm o złożoności  $O(n^2)$  może być lepszy od algorytmu o złożoności  $O(n \log n)$ ?
2. W jakich sytuacjach tablice haszujące są lepsze od drzew czerwono-czarnych?
3. Zaimplementować funkcję, która znajduje i usuwa największy element z listy dwukierunkowej. Zapisz potrzebne struktury.
4. Podaj reguły kolorowania drzew czerwono-czarnych. Pokoloruj drzewo:



5. Napisz na czym polega funkcjonalność kolejki priorytetowej. Podaj 3 algorytmy, w których jest stosowana wraz z krótkim opisem jaki problem rozwiązuje.
6. Jakich struktur danych użyć dla reprezentacji zbiorów rozłącznych, tak aby operacje FindSet i operacja wykonywania iloczynu zbiorów były jak najszybsze.
7. Kiedy reprezentacja grafu przez macierz adiacencji jest lepsza od reprezentacji przez listy sąsiadów?
8. Zaimplementuj DFS dla grafu w postaci macierzowej.
9. Podaj przykład algorytmu zachłannego i algorytmu dynamicznego. Krótko opisz zasadę działania jednego z nich.
10. Podaj ideę tablic haszujących.