

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).