

ASD – egzamin – II termin – VII.2015

Zadanie 1. (zadania krótkie) Proszę rozwiązać następujące zadania:

1. [5pkt.] Proszę zaimplementować możliwie jak najszybszą funkcję sortującą tablicę n liczb.

2. [5pkt.] Dany jest typ `struct List {List * next; int val; }` realizujący listę jednokierunkową. Proszę zaimplementować funkcję `List * reverse { List *L; }`, która odwraca kierunek listy L (bez wartownika) i zwraca wskaźnik na jej głowę.

[10 pkt.] **Zadanie 2.** Dany jest ważony, nieskierowany graf pełny G zawierający n wierzchołków, reprezentowany przez macierz wag krawędzi. Proszę zaimplementować funkcję

`int MST(int **G, int n)`

która otrzymuje na wejściu ten graf i zwraca sumę wag krawędzi minimalnego drzewa rozpinającego dla G . Funkcja powinna być możliwie jak najszybsza. Można założyć, że dostępna jest struktura:

`struct List { int u, v; int w; };`

oraz funkcja `List *sort { List *L }`, która sortuje niemalejąco podaną listę L według pól w (lista L zawiera wartownika i funkcja `sort` zwraca na niego wskaźnik). Wszystkie pozostałe funkcje i struktury należy zaimplementować.

[10 pkt.] **Zadanie 3.** Dana jest następująca struktura opisująca drzewo:

```
struct Tree{
    Tree *parent;      //rodzic, lub NULL jeśli to korzeń
    Tree *left, *right; //lewe i prawe dziecko
    int w_left, w_right //wagi krawędzi do lewego i do prawego dziecka (dodatnie)
}
```

Proszę opisać (bez implementacji) możliwie jak najszybszy algorytm, który mając na wejściu drzewo opisane przez strukturą `Tree` znajduje długość najdłuższej ścieżki między dwoma węzłami drzewa (prosta ścieżka to taka, która nie odwiedza żadnego węzła więcej niż raz). (Na potrzeby algorytmu mogą państwo uzupełnić strukturę `Tree` o dalsze pola.)