

## ALGORYTMY I STRUKTURY DANYCH

IIUWr. II rok informatyki.

1. (1pkt) Napisz w pseudokodzie procedury:

- przywracania porządku
- usuwania minimum
- usuwania maksimum

z kopca minimaxowego. Przyjmij, że elementy tego kopca pamiętane są w jednej tablicy (określ w jakiej kolejności). Użyj pseudokodu na takim samym poziomie szczegółowości, na jakim zostały napisane w Notatce nr 2 odpowiednie procedury dla zwykłego kopca.

2. (1,5pkt) Ułóż algorytm, który dla danych  $k$  uporządkowanych niemalejąco list  $L_1, \dots, L_k$  liczb całkowitych znajduje najmniejszą liczbę  $r$ , taką że w przedziale  $[a, a+r]$  znajduje się co najmniej jedna wartość z każdej z list  $L_i$ , dla pewnej liczby  $a$ .

Twój algorytm nie może modyfikować list  $L_i$  i powinien być pamięciowo oszczędny (no i oczywiście jak najszybszy).

3. (1pkt) *Porządkiem topologicznym* wierzchołków acyklicznego digrafu  $G = (V, E)$  nazywamy taki liniowy porządek jego wierzchołków, w którym początek każdej krawędzi występuje przed jej końcem. Jeśli wierzchołki z  $V$  utożsamimy z początkowymi liczbami naturalnymi, to każdy ich porządek liniowy można opisać permutacją liczb  $1, 2, \dots, |V|$ ; w szczególności pozwala to na porównywanie leksykograficzne porządków.

Ułóż algorytm, który dla danego acyklicznego digrafu znajduje pierwszy leksykograficznie porządek topologiczny.

4. (2pkt) *Nad słowem* zbioru słów  $\{w_1, w_2, \dots, w_k\}$  nazywamy dowolne słowo  $W$ , takie że  $\forall_{i=1, \dots, k}$   $w_i$  jest pod słowem słowa  $W$ . Ułóż algorytm, który dla zbioru słów dwuliterowych (tj. takich, że  $\forall_{i=1, \dots, k}$   $|w_i| = 2$ ) znajduje (jakiś) najkrótsze jego nad słowo.

5. (1pkt) Ułóż algorytm, który dla zadanego acyklicznego grafu skierowanego  $G$  znajduje długość najdłuższej drogi w  $G$ . Następnie zmodyfikuj swój algorytm tak, by wypisywał drogę o największej długości (jeśli jest kilka takich dróg, to Twój algorytm powinien wypisać dowolną z nich).