Lista nr 10 z matematyki dyskretnej

1. Przypuśćmy, że w grafie G wszystkie wagi krawędzi są różne. Pokaż, nie używając żadnego algorytmu, że G zawiera tylko jedno minimalne drzewo rozpinające.

2. Niech T będzie MST grafu G. Pokaż, że dla dowolnego cyklu C grafu G drzewo T nie zawiera jakiejś najcięższej krawędzi z C.

3. Czy poniższy algorytm zawsze znajduje MST w grafie spójnym G?

   Załóżmy, że krawędzie grafu są posortowane wg wag: $w(e_1) \leq w(e_2) \leq \ldots \leq w(e_m)$. Dla każdej krawędzi o indeksie i w kolejności od m do 1 wykonaj następujące: jeśli wyrzucenie e_i nie rozspaja G, wyrzuć e_i z G.

4. Udowodnij, że algorytm Prima znajdowania MST działa poprawnie.

5. Udowodnij, że algorytm Boruvki rzeczywiście znajduje drzewo rozpinające, tzn. pokaż, że w żadnej iteracji nie powstaje cykl.

6. Topologiczne porządkowanie wierzchołków acyklicznego digrafu. Niech D będzie digrafem acyklicznym, tzn. D nie zawiera cykli skierowanych. Podaj algorytm, który w czasie $O(m+n)$ porządkuje wierzchołki digrafu w taki sposób, że po uporządkowaniu, jeśli (i,j) jest krawędzią skierowaną w D, to $i < j$.

7. Digraf D (tj. graf skierowany) jest dany w postaci macierzy sąsiedztwa. Wykaż, że sprawdzenie, czy D zawiera źródło, czyli wierzchołek, z którego wychodzą krawędzie do wszystkich pozostałych wierzchołków, ale nie wchodzi do niego żadna krawędź, może być wykonane w czasie liniowym względem liczby wierzchołków w D. Zapisz swój algorytm w jakimś języku programowania i określ dokładnie jego złożoność obliczeniową, jako funkcję zmiennej liczby wierzchołków w digrafie.

Katarzyna Paluch