Struktury Danych i Złożoność Obliczeniowa – ćwiczenia

09.05.2018 – algorytmy grafowe część II

A) Algorytmy zachłanne
(R. Neapolitan..., "Podstawy algorytmów z przykładami w C++" (2004r.), rozdział 4 pp. 161 - 174)

a) definicje: problem, problem optymalizacyjny, problem decyzyjny

b) podstawy strategii zachłannej: własność wyboru zachłannego i optymalna podstruktura

c) problem plecakowy (ciągły (*0-1 knapsack problem*) i dyskretny) - wersja optymalizacyjna
(T. Cormen, "Wprowadzenie do algorytmów" (wydanie czwarte 2001r.) rozdział 17.3, pp. 381 - 383,
R. Neapolitan, K. Naimipour, "Podstawy algorytmów z przykładami w C++" (2004r.), pp. 194 - 201)

d) kompresja *Huffmana*: kody bezprefiksowe, drzewo *trie*, tworzenie drzewa *trie*, algorytm
(R. Sedgewick, "Algorytmy" (wydanie czwarte 2017r.) rozdział 5, pp. 838 - 851,
T. Cormen..., "Wprowadzenie do algorytmów" (wydanie czwarte 2001r.) rozdział 17.3, pp. 383 - 390,
A. Drozdek, "C++ Algorytmy i struktury danych" (2004r.) rozdział 11, pp. 503 - 518,
R. Neapolitan, K. Naimipour, "Podstawy algorytmów z przykładami w C++" (2004r.), pp. 188 - 194)

A_1) Zadania

ad. c) Dane są zbiór N elementów $\{x_1, x_j, ..., x_N\}$, każdy o określonej wartości s_j i wadze w_j oraz plecak o nośności B. Optymalnym rozwiązaniem jest, taki zbiór M, że: M jest podzbiorem N, suma wag w M jest nie większa niż B, suma wartości w M jest największa.

→ podać algorytm rozwiązania oraz pseudokod algorytmu

→ rozwiązać problem dla następujących danych:

i) $n = 3$, $[n_j, s_j, w_j]$ [1, 60, 10], [2, 100, 20], [3, 120, 30], $B = 50$

ii) $n = 5$, $[n_j, s_j, w_j]$ [1, 6, 2], [2, 5, 5], [3, 10, 3], [4, 12, 4], [5, 5, 4], $B = 16$

ad. d) Dany jest zbiór znaków {a, b, c, d, e}. Znaki występują (w tekście) odpowiednio:

iii) 23, 11, 5, 16, 2, 1

iv) 45, 13, 12, 16, 9, 5 (tysięcy razy).

Utworzyć zestaw kodów bezprefiksowych stosując algorytm *Huffmana*. Należy pokazać proces tworzenia drzewa *trie*.

v) dla tekstu *a man, a plan, a canal panama*. Utworzyć zestaw kodów bezprefiksowych stosując algorytm *Huffmana*. Należy pokazać proces tworzenia drzewa *trie*.

[jest jeszcze jedna strona ↓]

B) Minimalne drzewa rozpinające (*Minimum Spanning Tree - MST*) – idea działania, pseudokod, złożoność dla różnych reprezentacji grafów, ich wielkości czy gęstości.

(T. Cormen..., "Wprowadzenie do algorytmów" (wydanie czwarte 2001r.) rozdział 24, pp. 562 – 577,
R. Sedgewick, "Algorytmy" (wydanie czwarte 2017r.) rozdział 4.3, pp. 616 – 648,
R. Neapolitan..., "Podstawy algorytmów z przykładami w C++" (2004r.), rozdział 4.1 pp. 161 – 174)
(skorzystać z wizualizacji: https://www.cs.usfca.edu/~galles/visualization/Algorithms.html,
https://visualgo.net/en,
tomasz.kaplon.staff.iiar.pwr.wroc.pl SDiZO (EZI), *wykład*: **Algorytmy grafowe PDF**)

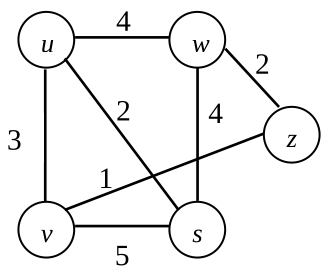
a) algorytm *Jarníka·Prima*

b) algorytm *Kruskala*

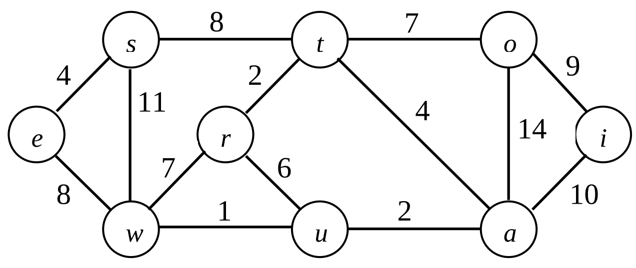
c) algorytm *Borůvki*

B$_1$) Zadania

Dla zadanych grafów G_4 (z poprzedniej listy)(potraktować jak nieskierowany), G_5, G_6, G_7 oraz G_8 wyznaczyć *MST* algorytmami *Jarníka·Prima*, *Kruskala*, *Borůvki*:



G_5



G_6

v_i		v_j, c_{ij}		v_j, c_{ij}		v_j, c_{ij}		v_j, c_{ij}
0	→	2, 3	→	4, 1				
1	→	2, 3	→	5, 2				
2	→	0, 3	→	1, 3	→	5, 6	→	6, 8
3	→	5, 3	→	7, 5				
4	→	0, 1	→	6, 1	→	7, 1		
5	→	1, 2	→	2, 6	→	3, 3	→	6, 9
6	→	2, 8	→	4, 1	→	5, 9	→	7, 9
7	→	3, 5	→	4, 1	→	6, 9		

G_7

https://github.com/georgicodes/algorithmics/blob/master/java/algs4-data/mediumEWG.txt ☺

G_8

[nie ma więcej stron]