

APCM și DRUMURI MINIME DE SURSĂ UNICĂ

Arbori parțiali de cost minim	2
Drumuri minime de sursa unică	3
Tema obligatorie (1,2 puncte – 0,2 fiecare problema)	4
Tema suplimentară (1,4 puncte – 0,2 fiecare problema).....	4

Arbori parțiali de cost minim

Fișierul `grafpond.in` are următoarea structură: numărul de vârfuri n , numărul de muchii m și lista muchiilor cu costul lor (o muchie fiind dată prin extremitățile sale și cost). Costul unei muchii este număr întreg.

grafpond.in
5 7
1 4 1
1 3 5
1 2 10
2 3 2
4 2 6
4 5 12
5 2 11

- a) Implementați algoritmul lui Kruskal pentru determinarea unui arbore parțial de cost minim al unui graf conex ponderat cu n vârfuri și m muchii. Graful se va citi din fișierul `grafpond.in`. $O(m \log n)$ (+ și versiunea $O(n^2 + m \log n)$)
<https://www.infoarena.ro/problema/apm>
b) Modificați programul de la a) astfel încât să determine (dacă există) un arbore parțial de cost cât mai mic care să conțină 3 muchii ale căror extremități se citesc de la tastatură. Se vor afișa muchiile arborelui determinat.
2. Implementați algoritmul lui Prim pentru determinarea unui arbore parțial de cost minim al unui graf conex ponderat cu n vârfuri și m muchii. Graful se va citi din fișierul `grafpond.in`. $O(m \log n)$ (+ și versiunea $O(n^2)$) <https://www.infoarena.ro/problema/apm>
3. **Second best minimum spanning tree** – Implementați un algoritm eficient pentru determinarea primilor doi arbori parțiali cu cele mai mici costuri, pentru un graf conex ponderat, în ipoteza că muchiile au costuri distincte ($O(n^2)$). Graful se va citi din fișierul `grafpond.in`.



Determinarea următorului arbore cu costul cel mai mic după arborele parțial de cost minim poate fi utilă spre exemplu ca soluție secundară în probleme care se reduc la determinarea unui apcm (conectare, revizie rețea etc)

grafpond.in	iesire
6 7	Primul
1 2 13	Cost 60
1 3 14	Muchii
2 3 16	1 2
1 4 11	1 3
1 5 12	1 4
4 5 15	1 5
5 6 10	5 6
	Al doilea
	Cost 62
	Muchii
	1 2
	2 3
	1 4
	1 5
	5 6

Drumuri minime de sursa unică

1. **Drum critic (Critical Path Method).** Se citesc din fișierul `activitati.in` următoarele informații despre activitățile care trebuie să se desfășoare în cadrul unui proiect:

- n – numărul de activități (activitățile sunt numerotate $1, \dots, n$)
- d_1, d_2, \dots, d_n durata fiecărei activități
- m – număr natural
- m perechi (i, j) cu semnificația: activitatea i trebuie să se încheie înainte să înceapă j

Activitățile se pot desfășura și în paralel.

- a) Să se determine timpul minim de finalizare a proiectului, știind că acesta începe la ora 0 (echivalent – să se determine durata proiectului) și o succesiune (critică) de activități care determină durata proiectului (un drum critic – v. curs) **$O(m + n)$** .
- b) Să se afișeze pentru fiecare activitate un interval posibil de desfășurare (!știind că activitățile se pot desfășura în paralel) **$O(m + n)$** .

activitati.in	iesire
6	Timp minim 47
7 4 30 12 2 5	Activitati critice: 4 3 6
6	1: 0 7
1 2	2: 7 11
2 3	3: 12 42
3 6	4: 0 12
4 3	5: 42 44
2 6	6: 42 47
3 5	

Robert Sedgewick and Kevin Wayne, *Algorithms, 4th Edition, Addison-Wesley, 2011*.

2. Se citesc din fișierul `grafpond.in` informații despre un graf **neorientat** ponderat și de la tastatură un număr k , o listă de k puncte de control ale grafului. Determinați cel mai apropiat punct de control de vârful s și afișați un lanț minim până la acesta, folosind algoritmul lui **Dijkstra** (+ problema 5 din laboratorul 1 pentru cazul ponderat) - **$O(m \log(n))$** .

Tema obligatorie (1,2 puncte – 0,2 fiecare problema)

Implementați algoritmi eficienți pentru rezolvarea următoarelor probleme. Se vor adăuga comentarii în cod cu ideea algoritmului și complexitatea sa.

- <http://www.infoarena.ro/problema/retea2>
- <https://www.infoarena.ro/problema/disjoint>
- **Clustering.** Fișierul cuvinte.in conține cuvinte pe mai multe linii, cuvintele de pe o linie fiind separate prin spațiu. Se citește de la tastatură un număr natural k. Se consideră distanța Levenshtein între două cuvinte https://en.wikipedia.org/wiki/Levenshtein_distance. Să se împartă cuvintele din fișier în k clase (categorii) nevide astfel încât gradul de separare al claselor să fie maxim (= distanța minimă între două cuvinte din clase diferite) - **v. curs**; se vor afișa pe câte o linie cuvintele din fiecare clasă și pe o altă linie gradul de separare al claselor.

cuvinte.in	Ieșire pentru k=3 (clasele nu sunt unice, dar gradul de separare da)
martian care este sinonim ana case apa arbore partial minim	care este ana case apa arbore martian partial sinonim minim 4

- <https://leetcode.com/problems/path-with-maximum-probability>
- <http://www.infoarena.ro/problema/catun>
- <https://infoarena.ro/problema/easygraph>

Tema suplimentară (1,4 puncte – 0,2 fiecare problema)

- <https://leetcode.com/problems/min-cost-to-connect-all-points>
- <https://leetcode.com/problems/checking-existence-of-edge-length-limited-paths>
- <https://www.infoarena.ro/problema/online>
- <https://infoarena.ro/problema/apm2>
- <https://www.infoarena.ro/problema/ciclu>
- <https://www.infoarena.ro/problema/lanterna>
- <https://leetcode.com/problems/cheapest-flights-within-k-stops>