1 noiembrie 2022

Tema 3
(seminar)

ALGORITMI FUNDAMENTALI

Tema Seminar 3

(Enescu Irina Ștefania 233)

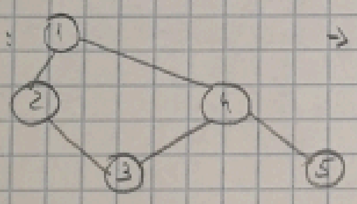
6) $G = (V,E)$ graf neorientat conex : $\begin{cases} V = \{1, 2, \dots n\} \\ \text{relație de ordine} < \text{pe } V \end{cases}$

Se cal. DFS și BFS ce pornesc din 1 și vizitează vecinii unui nod în ordinea <.
Ce prop. trb. să respecte G pt. ca DFS și BFS să aibă același arbore parțial $T_{DFS} = T_{BFS}$?

$T_{DFS} = T_{BFS} \Leftrightarrow \begin{cases} \text{au aceeași rădăcină (nodul 1)} \\ \text{au aceleași muchii, } |V|-1 \text{ la număr} \end{cases}$

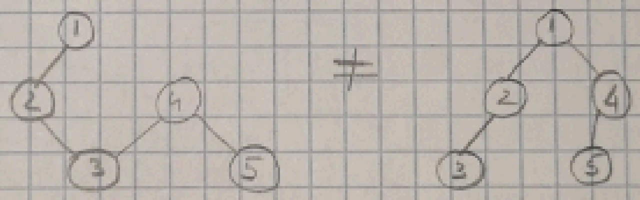
G neorientat conex

Exemplu:



-> avem un ciclu 1-2-3-4-1

DFS : 1 2 3 4 5   BFS : 1 2 4 3 5



Dacă avem un ciclu, toate nodurile din ciclul respectiv vor fi pe o parte a T_{DFS}, în timp ce în T_{BFS} vor fi despărțite. Cum graful este neorientat conex, toate nodurile vor forma un T_{DFS} sub formă de lanț, pe când T_{BFS} va avea structură de arbore. Așadar, ca $T_{DFS} = T_{BS}$, graful trebuie să fie un arbore.

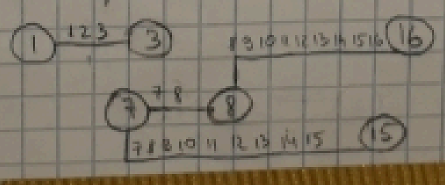
7) n intervale închise $[a_i, b_i]$, $1 \le i \le n$

Alegem submulțime S de interv. a.î. oricare două se intersect. în max. un punct, iar suma lung. de interv. e maximă. Cum?

- $(V,E) = ?$ Cum e graful?

Graful este neorientat $\begin{cases} \text{noduri} \rightarrow \text{nr. nat. din capetele interv.} \\ \text{muchii} \rightarrow \text{mulțimea nr. din interval (intervalul în sine)} \end{cases}$

ex.: $[7,15]$, $[8,16]$, $[1,3]$, $[7,8]$



• Cum rez. problema?

Pornim de la nodul cu val. cea mai mică din graf.

Pentru ca două interv. să se intersect. în max. un punct, fie nu se intersectează, fie au un capăt comun. Mergem în continuare în comp. conexă cu BFS salvând lungimea parcursă pentru a o alege pe cea mai mare la final. Salvăm acest drum.

Repetăm pasul anterior pentru fiecare componentă conexă.

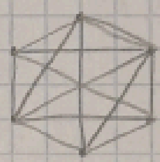
După ce avem toate drumurile max. din fiecare comp. conexă, le selectăm astfel încât să nu se intersecteze în mai mult de 1 punct folosindu-ne de informația din muchii.

Ne uităm la capetele fiecărui drum și elim. bucăți cât mai mici care sunt comune

8) $G = (V, E)$ graf neorientat $\quad V = \{1, 2, .. n\}$

$\theta(i,j)$ — 1 avem muchie i-j
— 0 nu avem muchie i-j

Pt. orice algoritm de det. a comp. conexe există un graf G_A a.î. A trebuie să facă cel puțin $\frac{(n-1)n}{2}$ interogări.



$n = 6$

$n = 5+4+3+2+1 = 15$

iar $15 = \frac{5(5+1)}{2}$

Obs: Un graf complet are $\frac{n(n-1)}{2}$ muchii.

Pentru fiecare nod, algoritmul va trebui să facă interogări pentru nodurile la care nu a ajuns. Pt. 1 face la 2, 3, 4, 5, 6; Pt. 2 face la 3, 4, 5, 6 deoarece 2-1 a fost deja făcut ş.a.m.d. $\Rightarrow (n-1)+(n-2)+\dots$

$\Rightarrow \frac{n(n-1)}{2}$ interogări minime.

9) n pct. de interes (x_i, y_i) $\quad\uparrow$ pleacă de la (x_i, y_i) și se întoarce tot aici, vizitând toate nodurile

Un turist realizează un ciclu prin noduri și vrea ca dist. să fie de cel mult două ori mai mare decât cea minimă. Cum face asta?

Facem un graf cu nodurile punctele de interes și muchiile toate distanțele posibile între noduri (un graf complet). Calculăm toate ciclurile posibile ce pornesc din (x_i, y_i) și vizitează tot, reținând valoarea minimă și ciclul corespunzător - complexitate mare :(

Facem același graf și un APM, apoi închidem ciclul cu distanța minimă cea mai apropiată de ultima vizită.