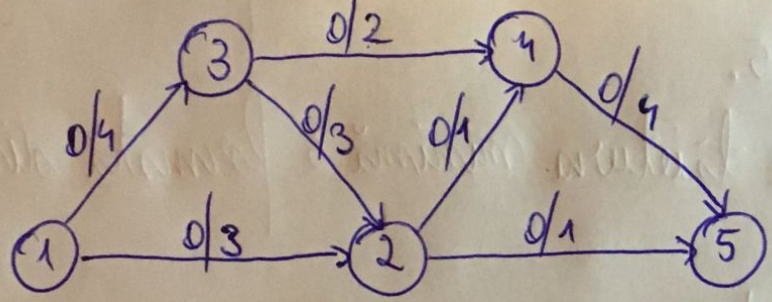
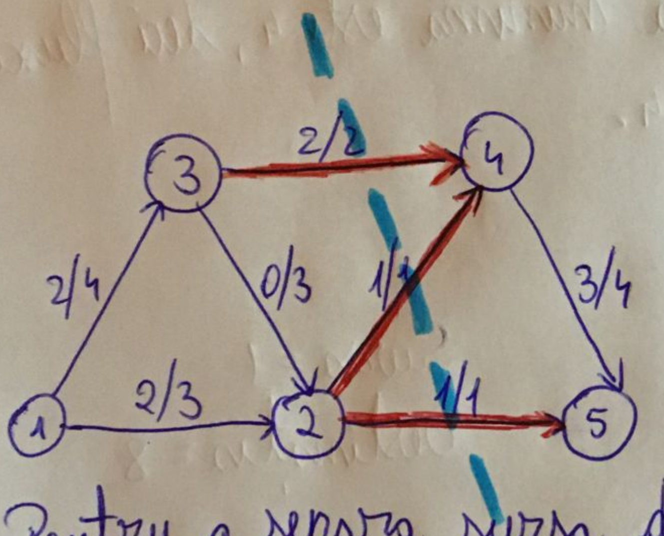
TEMA 2 SEMINAR
POPESCU PAULLO ROBERTTO KARLOSS
-DECEMBRIE 2021-

Exercițiul 1

Avem fluxul:



S = 1 (Sursa)
T = 5 (Destinația)



Fluxul maxim este 4.
Știm că fluxul maxim este egal cu tăietura minimă.

Pentru a separa sursa de destinatie, vom împărți nodurile în 2 multimi distincte. O tăietură este formată din muchiile ce despart cele două mulțimi. Căutăm tăietura minimă (cea cu costul cel mai mic)

- {1}, {2, 3, 4, 5} cu tăietura minimă 7 formată din muchiile 1-2, respectiv 1-3.
- {1, 2}, {3, 4, 5} cu tăietura minimă 6 formată

din muchiile 1-3, 2-4, 2-5.

• {1,2,3}, {4,5} cu tăietura minimă 4 formată din muchiile 3-4, 2-4, 2-5.

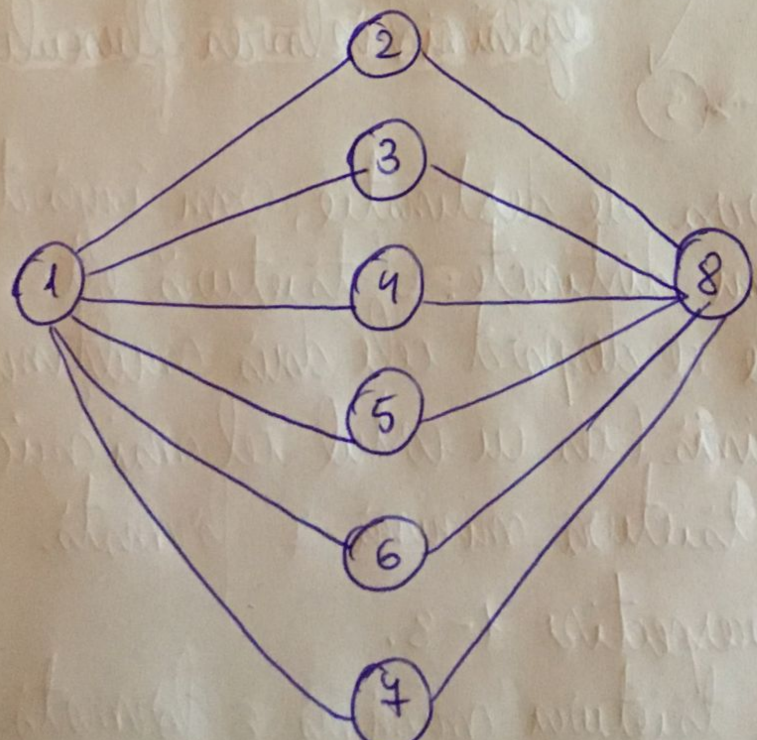
• {1,3}, {2,4,5} cu tăietura minimă 8 formată din muchiile 1-2, 3-2, 3-4.

• {1,3,4}, {2,5} cu tăietura minimă 10 formată din muchiile 1-2, 3-2, 4-5.

• {1,2,3,4}, {5} cu tăietura minimă 5 formată din muchiile 2-5, 4-5.

Observăm că tăietura minimă este 4, deci fluxul maxim este într-adevăr 4.

Exercițiul 2



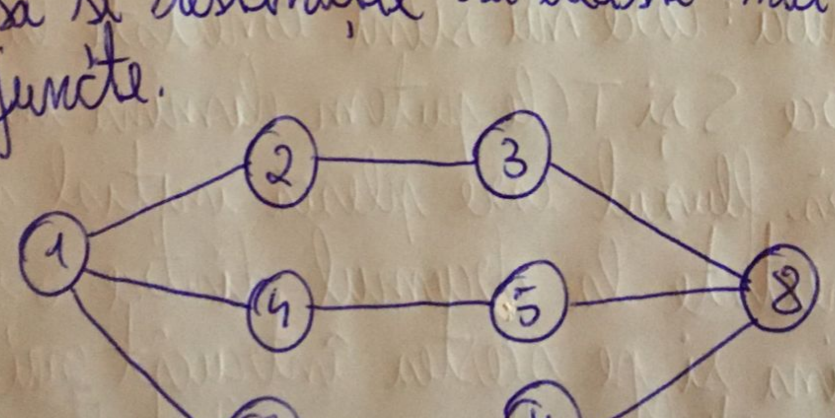
Sursa = 1
Destinația = 8

Algoritmul lui Edmonds-Karp caută cele mai scurte drumuri către destinații (8 în exemplul nostru). Cât timp găsește un drum pe care mai fi transmis flux, determină cantitatea de flux care mai poate fi adăugată, și o adaugă după aceea. În acest graf, algoritmul Edmonds-Karp poate face 8 parcurgeri BFS, în funcție de implementare.

Dacă generalizăm acest graf, la unul cu n noduri, dintre care unul este sursa, iar celălalt este destinația, restul de n-2 noduri vor fi conectate doar la sursă și la destinație. Așadar Edmons-Karp va face O(V) parcurgeri BFS, fiecare parcurgere având complexitatea O(V+E).

$\Rightarrow$ Complexitatea este O(V·E)

Dacă mai generalizăm acest graf a. î. între sursă și destinație să existe mai multe lanțuri disjuncte.



Avem k lanțuri, deci alg. va face aprox k parcurgeri BFS.

$k(V+E) < 2E^2$ pași, unde $k < V < E$.

În concluzie, algoritmul Edmonds-Karp, care determină fluxul maxim are complexitatea $\Omega(E^2)$ WORST-CASE!

Exercițiul 3

Regulă pentru orice nod, flux intrare = flux ieșire. Știm că dacă într-o rețea există muchii care pleacă din destinația T, pe acestea nu se poate duce mai mult flux decât a intrat. Acestea nu pot face decât ca fluxul să rămână egal (dacă avem un ciclu în care muchiile se întorc în rețea) sau să scadă. Ele nu influențează fluxul maxim deci le putem elimina.

Fluxul care pleacă din sursa S, poate fi maxim fluxul muchiilor care pleacă din S. Dacă într-o rețea există muchii care ajung în S, acestea pot fi de două tipuri: care nu sunt conectate la drumul dintre S și T (le putem elimina întrucât nu influențează fluxul care pleacă inițial din S), care sunt conectate la drumul dintre S și T (le putem elimina și pe acestea întrucât nu se poate întoarce decât fluxul care a plecat din S, ar intra într-o buclă și nu ar influența fluxul maxim).

A se întoarce flux pe muchiile care ajung în destinația T este echivalent cu a exista o muchie care pleacă din T, doar că aceasta este de întoarcere. Așadar, după cum a precizat anterior, acest lucru nu este posibil. A se întoarce flux pe muchiile care pornesc din sursa S, este echivalent cu a exista muchie care ajunge în S, doar că aceasta este de întoarcere. Deci, acest lucru nu este posibil.

Exercițiul 4

În plus față de problema clasică de flux care muchiile u - v au capacitatea $c(u, v)$, în problema noastră și nodurile au capacitate. Putem reduce această problemă la cea anterioară transformând constrângerea la nivel de noduri în una la nivel de muchii. Orice nod v va fi înlocuit cu nodurile v' și v'', între care există muchie orientată de la v' la v'', cu costurile $c(v', v'') = c(v)$ și $c(v') = c(v'') = 0$. În nodul v' vor intra muchiile care intră în v, iar din nodul v'' vor ieși muchiile care ies din v. Pe această rețea, în care muchiile au cost, iar nodurile nu, putem aplica un algoritm de flux maxim.

În concluzie, putem calcula fluxul maxim folosind algoritmul Edmonds-Karp ($\Omega(V^2)$).

Exercițiul 5

Abordăm problema ca una de flux maxim. Pornim de la o sursă S și o destinație T. Vom avea câte un nod, x_i, pentru fiecare nod din arbore. Fiecare nod înafară de rădăcină, din arbore are un părinte, p_i. Pentru fiecare oraș avem s_i oameni și d_i locuințe. La început, vom adăuga, pentru fiecare nod x_i, 2 muchii: muchia $S - x_i$ cu capacitatea s_i și muchia $x_i - T$ cu capacitatea d_i.

Noi trimitem un număr de oameni (flux care pleacă din S) și cazăm un număr de oameni (fluxul care intră în T). Momentan, putem găsi o locuință pentru un om doar în orașul său, dar un om poate locui în orice oraș (în sus pe arbore, până la rădăcină). Așadar trebuie să mai adăugăm și muchia $x_i - p_i$. Această muchie nu va avea nicio capacitate maximă. Putem trimite oricâți oameni dintr-un oraș într-un oraș părinte, mai bine spus, pentru că fiecare nod are o astfel de muchie, putem trimite oricâți oameni dintr-un oraș (nod) în alt oraș din drumul spre capitală (RĂDĂCINA ARBORELUI), inclusiv în aceasta.

În acest punct, putem afla dacă există o strategie de relocare prin care toți oamenii să aibă o locuință. Fluxul maxim în rețea va reprezenta numărul de oameni care și-au găsit o locuință. Dacă la final, fluxul maxim va fi egal cu numărul de oameni din toate orașele, atunci toți și-au găsit locuința, altfel există câțiva pe care i-am trimis să își caute locuința și pe care nu i-am cazat, ei au pornit din sursa dar nu au reușit să ajungă în destinație.

Pentru fiecare oraș este normal să se cazeze cât mai mulți locuitori. Începând din frunze, vom satura fiecare oraș cu minimul dintre numărul de locuitori și numărul de locuințe. Mai exact, vom trimite cât de mult de flux putem pe muchia $x_i - T$.

Dacă numărul de locuitori e mai mic ca numărul de locuințe, ne-am terminat treaba cu orașul curent. Îl putem șterge direct și adăugăm fluxul curent la fluxul maxim.

Dacă numărul de locuitori e mai mare decât capacitatea orașului de a-i caza, va trebui să îi trimitem în orașul părinte. Adăugăm fluxul pe care l-am putut trimite la fluxul maxim și continuăm.

După cum am spus, locuitorii care nu a fost cazați vor fi trimiși în orașul părinte. Asta înseamnă că fluxul rămas (diferența dintre d_i și s_i) va fi trimis pe muchia $x_i - p_i$. Putem să ștergem muchia $x_i - p_i$ și să trimitem fluxul (diferența dintre d_i și s_i) direct din S în p_i. Acum putem să ștergem nodul x_i.

Vom face acest lucru începând cu frunzele și vom șterge, pe rând, câte un nod, manipulând fluxul după cum am precizat mai sus. La sfârșit vom rămâne doar cu sursa S și destinația T, rezultatul fiind fluxul maxim dintre cele 2 noduri.

Ștergerea unui nod $O(1)$, ștergem n noduri. Calcularea fluxului se va face în $O(1)$, total $O(n)$.