Zadanie úloh z predmetu Algoritmická teória grafov na týždeň 30.3. - 3.4.2020

Vážení študenti predmetu Algoritmická teória grafov. Karanténa pokračuje a zdá sa, že dištančná forma výučby bude pokračovať do konca semestra. Skúšky bude možné urobiť tiež len na diaľku. Za tejto situácie bude podstatné vypracovanie všetkých zadaných algoritmov a ich prezentácia cvičiacim. To môže zvádzať kopírovať si algoritmy jeden od druhého.

Dôrazne upozorňujem, že pri nájdenej zhode obaja predkladatelia skončia výsledkom FX a podaním na disciplinárne konenie za pokus o povod.

O otázkach ku skúške platí všetko, čo som uviedol v „Zadaní úloh z predmetu Algoritmická teória grafov na týždeň 23. - 27.4.2020“.

Stále platí, že mimoriadnu váhu pri skúške budem klásť na programy, ktoré vám priebežne zadávam.

Dôležité upozornenie

Pokúsim sa naštartovať videokonferenciu v regulárnom čase prednášky ATG, t. j. vo štvrtok o 10:00 hodine. Na videokonferenciu sa prihlásite pomocou prihlasovacích údajov, ktoré dostanete v maili tak 30 – 40 minút pred začatím konferencie. Netreba pre tým nič inštalovať. Ak systém po kliknutí na zadaný link vyžiada vaše meno, prosím o jeho korektné zadanie na zistenie účasti. Zatiaľ s tým vôbec neviem pracovať a ani neviem, či sa mi to podarí. V prípade, že áno, nezapínajte ani kamery, ani mikrofóny.

Na týždeň od 30.3. do 3.4.2020 zadávam tieto úlohy:

V rámci nahradenia prednášky:

Doštudujete do konca kapitolu 4 „Acyklické grafy, stromy a kostry“, časť 4.4   (str. 119) v učebných textoch:

https://frcatel.fri.uniza.sk/users/paluch/grafy.pdf

Príslušné slajdy nájdete tu:

https://frcatel.fri.uniza.sk/users/paluch/Prezentacie/GrafPrez_04.pdf

od trany 95 podľa číslovania Acrobat Reader, resp. od slajdu 16 do konca kapitoly.

Na doštudovanie nemáte toho veľa, venujte sa hlavne programovaniu.

Cvičenia:

Dokončite všetky zadané programy včítani Tarryho algoritmu. Programy robte tak, aby sa počítaný graf alebo digraf zadával menom súboru, odkiaľ sa načíta.

Naprogramujte Kruskalov algoritmus II.

POZOR. Algoritmy na hľadanie najkratšej cesty sme mali formulované pre **digrafy.** Ak sme potrebovali nájsť najkratšie cesty v grafe, robili sme to v (pomocnom) digrafe, ktorý mal pre každú

neorientovanú hranu {u,v} pôvodného grafu dvojicu orientovaných hrán (u,v), (v,u) s tou istou cenou.
 Ak sme počítali najkratšie cestu pre graf, mali sme v poli H[ ][ ] pre kažú hranu {u,v} dva riadky:

H[i][0], H[i][1], H[i][2],   H[j][0],H[j][1],H[j][2],

kde H[i][0]=u, H[i][1]=v,   H[j][0]=v, H[j][1]=u   a   H[i][2]=H[j][2]=cena hrany {u,v}

Tarryho algoritmus je určený pre **neorientované grafy**. Reprezentácia poľom H[ ][ ] použitá pre hľadanie najkratších ciest sa tu ukázala byť výhodná. Dva riadky poľa H[ ][ ] pre každú hranu umožnili vyjadriť, ktorým smerom bola hrana prejdená.

Kruskalov algoritmus je určený pre **neorientované grafy**.

Tento algoritmus nájdete na slajde 14 tu:
https://frcatel.fri.uniza.sk/users/paluch/Prezentacie/GrafPrez_04.pdf

a tiež v učebných textoch na str. 117:
https://frcatel.fri.uniza.sk/users/paluch/grafy.pdf

Skopírovaný algoritmus zo st. 14

Kruskalov algoritmus II. na hľadanie najlacnejšej (najdrahšej) kostry súvislého hranovo ohodnoteného grafu G = (V, H, c).
Krok 1. Zoraď' hrany podľa ich ohodnotenia vzostupne (zostupne) do postupnosti P.
Krok 2. Pre každý vrchol $i \in V$ polož $k(i) = i$.
Krok 3. Nech prvá hrana v postupnosti P je hrana {u, v}. Vylúč hranu {u, v} z postupnosti P.
Ak k(u) # k(v), zarad' hranu {u, v} do kostry, a
$\forall i \in V$, pre ktoré $k(i) = k(v)$, polož $k(i) := k(u)$
Krok 4. Ak je počet vybraných hrán rovný $|V| - 1$ alebo ak je postupnost' P prázdna, STOP.
Inak GOTO krok 3.

Na tento algoritmus môžete znova využiť pole H[ ][ ].
Tu ale stačí, aby pre každú hranu {u,v} skúmaného grafu bol v poli H[ ][ ] jednen riadok
H[i][0],H[i][1],H[i][2], kde H[i][0] = u, H[i][1] = v,   alebo H[i][0] = v, H[i][1] = u ,
t. j. nezáleží na poradí u a v.

Takýto súbor nájdete ako súbor pr3.hrn
https://frcatel.fri.uniza.sk/users/paluch/zadanie3.html

Súbor je zotriedený neklesajúco podľa prveho stĺpca.
```
Každá hranu {u,v} je uvedená len raz.
Ak u<v  a hrane zodpovedá riadok i,  potom H[i][0]=u, H[i][1]=v.
Ak u>v  a hrane zodpovedá riadok i,  potom H[i][0]=v, H[i][1]=u.

Počet hrán a počet vrcholov je nutné zistiť programovo.
Počet hrán zistite postupným načítavaním riadkov súboru pr3.hrn
ako stringov od začiatku až po EOF. Načítané stringy neukladajte.

Pocet úspešne načítaných reťazcov je počet hrán grafu.

Počet vrcholov je maximum čísel z prvého a druhího stĺpca súboru pr3.hrn.
POZOR!! Na rozdiel od 1. príkladu treba počítať maximum z oboch stĺpcov!!

Táto reprezentácia je vhodná napr. pre Kruskalov algoritmus II.
```

Návod na Kruskalov algoritmus II. pre hľadanie najlacnejšej kostry

Program si vypýta súbor s počítaným grafom – tuto pr3.hrn.
Zistite n – počet vrcholov grafu a m – počet hrán grafu v súbore zadanom z klávesnice
Dimenzujte pole H[ ][ ] tak aby malo tri stĺpce
Načítajte počítaný graf zo súboru do poľa H[ ][ ]
Usporiadajte pole H[ ][ ] neklesajúco podľa stĺpca 2. Teraz pole H[ ][ ] má zoradené hrany podľa ceny neklesajúco
Netreba počítať ple smerníkov S[]
Pre i=1 až i=n (počet vrcholov) položte K[i]=i
PočetHránKostry položte rovné 0.
Pre i=1 až i=m (počet hrán gfrafu) Zoberte hranu {u,v}, kde u= H[i][0], v= H[i][1].
Ak K[u] = K[v], choďte na ďalšiu hranu (continue)
Ak K[u] # K[v] zaraďte hranu i do kostry , zvýš PočetHránKostry o 1
a pre všetky vrcholy j od 1 do n urobte: Ak K[j] = K[v], polož K[j] = K[u].
AK PočetHránKostry = n – 1, **STOP**

Hrany kostry ukladajte buď do nejakého jednorozmerného poľa ako ich indexy v poli H, alebo do poľa s dvoma stĺpcami ako dvojice koncových vrcholov hrán kostry.

Pre menšie príklady navrhnite výpis nájdenej kostry.

Porozmýšľajte, čo sa stane, ako zadaný graf nie je súvislý.

Prémiová úloha.
Zistite všetky mosty daného grafu.
Návod. Ak je najká hrana {u,v} mostom, potom musí byť v každej kostre grafu.
Ak teda zostrojím akúkoľvek kostru grafu, mám n-1 podozrivých hrán – žiadna ďalšia už nemôže byť mostom.
Ako zistím, či hrana {u,v} už nájdenej najlacnejšej kostry je mostom?
Spustím znovu hľadanie najlacnejšej kostry s tým, že okrem hrán s rovnakými značkami K[] koncových vrcholov, zakážem aj hranu {u,v}.
Ak dostanem znovu kostru grafu bez hrany {u,v} (samozrejme už nemusí byť najlacnejšia, lebo jej chýba hrana {u,v}) , potom táto hrana nie je mostom.
Ak algoritmus skončí tak, že nájde menej ako n-1 hrán, potom graf bez hrany {u,v} nie je súvislý, a tak hrana {u,v} je mostom.
Výhodou tohto postupu je, že nemusím znovu usporiadavať pole H.

Modifikujte tento postup na hľadanie artikulácií.

Poznámka: Pri hľadaní mostov týmto spôsobom vôbec nemusím usporiadať pole H, pretože tu na cene vôbec nezáleží. Môžeme to tak robiť aj pre grafy, ktoré nie sú hranovo ohodnotené.

Predpokladám, že už máte naprogramované úlohy 1, 2, 3 a 4 zo zadania od 11. do 20.3.2020
a tiež Tarryho algoritmus. Ak nie, hrozí neúspech.

Pozdravuje
Stanislav Palúch