

V tejto úlohe potrebujeme vyrábať minimálnu kostru grafu a to pre každú hranu tak, aby obsahovala minimálna kostra grafu tú hranu. Preto najskôr spravme nasledovné: Vyrábajme si nalicnejšiu kostru grafu bez nijakých iných podmienok. Vieme to spraviť napr. Primovým algoritmom v čase $O(n^2)$. Potom vieme, že pre každú hranu, ktorú táto minimálna kostra grafu obsahuje je toto kostra, ktorú hľadáme. Ako však vyrábať kostru, pre ostatné hrany ?? Skúsme pridať teda do minimálnej kostry hranu, ktorú tam chceme mať. Ak sa táto hrana nenachádzala v kostre, tak v našom grafe minimálnej kostry ktorá je v tvare stromu vznikne kružnica. Minimálna kostra grafu s použitím tej hrany, ktorú sme tam pridali bude potom taký graf, keď z tej kružnice odstránime maximálnu hranu. Vieme, že táto kostra bude určite minimálna, pretože väčší prvok sme odstrániť už nemohli, takže určite je minimálna. Ako však v rozumnom čase nájsť maximálny prvok v kružnici ktorá nám vznikla ?? No ja teraz ukážem, že zistenie maximálneho prvku ide aj v konštantnom čase.

Zakoreňme si minimálnu kostru do nejakého vrchola r . Potom DFS prehľadávajme tento graf a pre každý vrchol v kde prideme(okrem koreňa) pridajme hranu v, r . vieme, že kružnica, ktorú sme vytvorili je vlastne postupnosť hrán po ktorých sme prišli z koreňa + hrana ktorú sme pridali. Ak prechádzaním DFS si pamätáme maximálnu hranu ktorú sme zatiaľ videli, tak v danom vrchole vieme zistiť maximálnu hranu ako $\max(\text{maximálnu hranu z vrchola o úroveň vyššie, a hranu ktorou sme sa tu z toho vrchola dostali})$. Ak si to vždy zapamätáme ktorá je pre daný vrchol maximálna hrana, tak potom pre všetkých synov daného vrchola vieme v konštantnom čase určiť maximálnu hranu.

Takto zakoreníme v každom vrchole a prejdeme vždy celý strom - takže určite prejdeme aj každú hranu - tj dokopy vyrábame v čase $O(n^2)$ minimálnu kostru pre každú hranu i, j .

Pamät: potrebujem si zapamätať hranu pre každé i, j a ešte aj minimálnu kostru pre každú hranu i, j . Dokopy to bude $O(n^2)$ pamäte. Program dá vždy správny výsledok, pretože z minimálnej kostry viem vyhodeniť maximálnej hrany z kružnice spraviť minimálnu kostru s použitím hrany i, j . Takže kostra je určite minimálna.

pseudokod:

Spravíme minimálnu kostru za použitia Primovho algoritmu, ktorý znie:

Všetky vrcholy si označme ako nenavštívené.

nastavme si pre každý vrchol vzdialenosť nekonečno.

Označme nejaký vrchol ako v .

Opakuj toľko krát, koľko je vrcholov:

Označme vrchol v ako navštívený.

Pre každý vrchol w , ktorý susedí s vrcholom v a ešte nieje navštívený nastavme vzdialenosť k nemu na $\min(\text{pôvodná vzdialenosť k } w, \text{vzdialenosť od vrchola } v \text{ k } w)$

Potom nájdime minimálnu vzdialenosť z vrchola v , ktorý ešte nieje navštívený.

Túto hranu pridajme do minimálnej kostry grafu, prirátajme dĺžku hrany k dĺžke kostry l a nastavme w ako vrchol v a opakujme znovu.

Pre každý vrchol v zakoreň graf vo vrchole v a rob DFS.

dfs funkcia(vrchol v , predok p , maxhrana)

nastav súčet minkostry na súčet $l + (\text{hrana } r, v) - \text{maxhrana}$

pre každého suseda vrcholu v ktorý nieje predok rob dfs(novyvrchol, vrchol v , max(maxhrana, hrana (vrchol v , novyvrchol)))