

Túto úlohu vieme riešiť viacerými spôsobmi. Prvý a najjednoduchší spôsob je, že vyskúšame všetky kombinácie zátarás. Pre každú kombináciu zistíme, či tvorí súvislú zátarasu a ak ano, tak či je najmenšia. Toto riešenie je však exponenciálne a my poznáme aj lepšie riešenie. Lepšie riešenie je v kvadratickom čase a použijem ho ako návod k môjmu vzorovému riešeniu. Určíme si smer, ktorým budeme prehliadať zátarasy. Nech je to BUNV v kladnom smere - tj najskôr prehliadneme zátarasu, ktorá začína na menšom čísle a až potom zátarasu ktoré začína na väčšom čísle. Potom pre každú zátarasu z nájdime takú zátarasu v , ktorá tvorí so z jednu súvislú zátarasu a končí čo najneskôr. To znamená, že keby sme pridali k zátarase z ľubovoľnú inú zátarasu, nikdy by tieto zátarasy netvorili oblúk, ktorý končí čo najneskôr. Označme takúto zátarasu v ako najlepšou zátarasou pre zátarasu z . Teraz chceme vypočítať takúto najlepšiu zátarasu pre každú zátarasu i . Tak v prvom rade si odstránime každú zátarasu i takú, že existuje taká zátarasa j , že celá zátarasa i leží v j - teda všetky úseky, ktoré obsahuje i obsahuje aj j . Prečo by sme mali každú takúto zátarasu i odstrániť ?? no keby mala byť v riešení, kde by sme použili najmenší počet zátarás, tak kludne by sme mohli použiť zátarasu j namiesto i . Takže zátarasu i vôbec nepotrebuje.

Potom si usortíme zvyšné intervaly podľa konca zostupne a prejdeme v takomto poradí všetky usortené zátarasy a pre každú viem v logaritmickom čase zistiť najlepšiu zátarasu. To môžeme spraviť tak, že si pamätám všetky zátarasy, ktoré som už prešiel a majú začiatok aspoň ako koniec zátarasy, ktorú teraz prehliadavame. Ostatné môžeme vymazať. A teraz nájdeme zátarasu, ktorá má najväčší koniec, pretože všetky zátarasy, ktoré si teraz pamätáme určite tvoria z so zátarasou, ktorú teraz prehliadavame jeden celok. Toto si môžeme pamätať v nejakej mape. Každú zátarasu do mapy pridáme práve raz, odstránime najviac raz a zisťujeme najlepšiu zátarasu práve raz, takže keď mapa má logaritmickú zložitosť, tak aj čas pre všetky zátarasy dokopy bude $n \cdot \log(n)$. Tu nám môže nastať ešte jeden problém, taký, že sme v kruhu, takže pre nejakú zátarasu sme nemuseli nájsť najlepšiu - pretože bola na konci a my sme začali od začiatku. Takže my to môžeme okášľať tak, že zátarasy prehliadneme dvakrát - dve celé kolieska - tj pre každú zátarasu určite budeme mať v mape takú zátarasu, že je jej najlepšia.

Keď sa teraz pozrieme na to, čo nám vzniklo zistíme, že je to orientovaný graf - každá hrana odkazuje na nejakú. Vieme, že záplata ktorá bude mať najmenej zátarás bude pozostávať len z najlepších zátarás. My však nevieme, kde začať, tak začneme na každej možnej zátarase a postupujeme po najlepších zátarasách až kým nevytvoríme kruh, okolo jazierka. Vieme, že optimálne riešenie určite bude začínať na jednej zátarase, takže my ich všetky vyskúšame. Toto nám však trvá n^2 času.

Toto riešenie však môžeme ešte vylepšiť. Keď sa pozrieme na orientovaný graf, ktorý nám vznikol, tak vidíme, že pozostáva z niekoľkých komponentov. Každý komponent pozostáva z práve jednej kružnice a z tej kružnice vyrastajú stromy. Takže môžeme prehľadávať každý komponent odstraňovaním listov, až pokým nám nevznikne len samotná kružnica. Stromy, ktoré sme prehliadavali, prehliadavame odstraňovaním listov a pre každý vrchol, keď je už list si pamätáme vrcholy, ktoré ešte nemajú celé koliesko okolo jazera. Aby sme to robili efektívne, tak pre každý vrchol - list si tieto vrcholy musíme pamätať v mape - alebo niečom podobnom, aby sme v tom mohli pekne vyhľadávať a listy, ktoré boli pod týmto vrcholom - pripojíme k tomuto vrcholu klasickým mergovaním stromov - menší pripojíme k väčšiemu. Ale keďže si to pamätáme v mape, tak čas budeme mať $n \cdot \log(n) \cdot \log(n)$, pretože mergovanie stromov má amortizovane logaritmický čas - každý vrchol pripojíme max $\log(n)$ krát. Potom nám v každom komponente zostane už len kružnica, ktorú spracujeme podobne, akurát si vyberieme ľubovoľný vrchol odkiaľ začať. Určite sme prešli každé koliesko okolo jazera - tj začali sme v každom vrchole, takže správnosť riešenia sa nám nezmenila. Mimochodom toto som aj submitol a zbehlo.

Ja som si však povedal, že toto riešenie pôjde aj vylepšiť aby bežalo v $n \cdot \log(n)$ čase. To znamená, že mergovanie stromov musíme vyhodiť, alebo spraviť v konštantnom čase. Spraviť v konštantnom čase to asi nepôjde, takže ich musíme vyhodiť. Ke ich vyhodíme, zostanú nám len kružnice. Bude najlepšie riešenie ležať na kružnici ?? Áno bude tam ležať. A to preto, lebo ak by sme začali niekde inde ako na kružnici, tak vidíme, že začiatočná zátarasa nieje najlepšou tej, ktorou končí táto postupnosť. Tak zoberme najlepšiu konca postupnosti a začiatočnú vymažeme. takto sa posúvame, až sa dostaneme s celou postupnosťou zátarás na kružnicu, takže určite bude ležať celá postupnosť zátarás na kružnici.

Takže stačí prehliadnúť kružnice a máme najlepšie riešenie. Vypísať ho už vieme jednoducho, keď sme si našli najlepšiu postupnosť - aspoň jej začiatok, pretože sa bude určite skladať s najlepších zátaras. Takže pre zažú zátarasu vieme určiť jej nasledovníka.

Čas riešenia je $n \cdot \log(n)$ a pamäť je lineárna - $O(n)$.

Pseudokod:

Usoříme všetky intervaly podľa konca a vymažeme všetky ktoré sa nachádzajú vo vnútri nejakých. Pre každý interval i nájdeme najlepší interval j

Spravíme si graf - pre každý vrchol si zapamätáme koho má ako predka a koľko má potomkov - aby sme vedeli či už je list, alebo nie

Pokiaľ nám nezostane žiaden list, tak odstraňujeme listy.

Keď už žiaden list nemáme, tak nám zostali už len kružnice. Pre každý vrchol ak ešte nebol navštívený - nachádza sa v kružnici pokračujeme po predkoch až pokým sa nám neminú všetky prvky v mape - zoznam tých vrcholov, ktoré ešte nemajú koliesko okolo jazierka. Ak nájdeme nejaké koliesko, tak ak má najmenej zátaras, tak si ho označíme.

Najlepšie koliesko vypíšeme.