

Riešenie ktoré je určite funkčné je, že prejdeme každé políčko mriežky a ak nájdeme všu, tak vyplníme štvorec ktorý vytvorí táto vša flakmi.

Takéto riešenie je ale veľmi pomalé a dá sa to spraviť rýchlejšie.

Prejdeme celú mriežku a pre každé políčko ktoré nájdeme a bude na ňom vša, tak zapíšeme si kde bude roh štvorca ktorý vytvorí táto vša (BUNV Ľavý horný).

Avšak tento roh môže byť aj mimo mriežky. Takže musíme všetko posunúť v poli, aby sa to zmestilo. Stačí nám to posunúť o n dole aj doprava, pretože najmenšie možné súradnice potom budú 0,0.

Mriežka sa potom bude nachádzať od n až po $n+s$ a riadky od n po $n+r$.

Toto nám zaberie $r*s$ času, pretože pre každé políčko zadané na vstupe spravíme konštantný počet operácií.

Pamäť pri tomto bude $O((n+r)*(n+s))$.

Pamäť sa však dá zredukovať aj na $O(r*s)$ ktorú popíšem neskôr.

Potom vieme, že pod každým rohom budú začínať riadky toho štvorca. Bude ich presne $2*n+1$. Preto tam poďme zapísať začiatky tých riadkov. Čo sa však stane ak zapisujeme – smerom zhora nadol a nájdeme ďalší roh ?? tak vieme, že začiatky riadkov rohu, ktoré zapisujeme a začiatku riadkov, ktoré prislúchajú štvorcu, ktorý je daný nájdeným rohom, tak niektoré začiatky riadkov sa budú prekrývať. Ale skôr ako skončia začiatky riadkov novo nájdeného rohu, tak skončia aj začiatky riadkov rohu, ktoré zapisujeme. Takže začiatky riadkov starého rohu môžeme ukončiť a zapisovať iba začiatky riadkov nového rohu.

Nejako takto:

```
for i=1 to n+s+1           // prejdeme všetky stĺpce
begin

koniec=-987654321          //nastavíme, že nezapisujeme konce
for j=1 to n+r+1           // pre každé políčko v stĺpci
begin
if mriezka[j][i]==roh then koniec=j+2*n+1 //ak je to roh tak označuj ďalších n*2+1 políčok ako
začiatky riadkov

if koniec>=j then označ políčko ako začiatok riadku      // ak máme značiť, tak značíme
end
end
```

keďže prejdeme každé políčko iba raz, tak čas bude $O((r+n)*(s+n))$

Teraz už máme začiatok každého riadku už len dorobiť všetky riadky.

Takže prejdeme každý riadok a ak nájdeme začiatok riadku štvorca, tak značíme políčka ktoré znečistí vša. Od začiatku riadku značíme $n*2+1$ políčok ako škvryny. Takže robíme niečo podobné ako z rohov na začiatky riadkov, ale teraz budeme prechádzať riadky a ak nájdeme začiatok riadku, tak značíme $n*2+1$ políčok ako škvryny. Ak nájdeme nový začiatok riadku, tak len predĺžime koniec.

Toto nám bude trvať $O(r*s)$ času, pretože stačí prezrieť riadky od 1 po r a stĺpce 1 až s , takže $r*s$.

Potom už len musíme vypísať oblasť mriežky, čo nám bude trvať $O(r*s)$ času.

Riešenie určite vždy dá správny výsledok, pretože do mriežky zapíšeme všetky flaky, ktoré tvorí a tým pádom ju vieme vždy vypísať tak ako má byť.

Ja som však sľúbil riešenie v pamäti $O(r*s)$

To je riešenie, ktoré mám submitnuté ako posledné – trošku zložitejšie, ale pamäť je $O(r*s)$.

Princíp je nasledovný:

Robíme všetko ako aj v predošlom riešení označujeme rohy, ale ak chceme zredukovať pamäť, tak potom niektoré rohy by boli mimo poľa. Čo v takomto prípade ??

NO ja vždy označím taký roh, ktorý sa nachádza v poli a označím ktorý to je.

Máme 4 možnosti – 4 rohy.

Ak sa ani jeden roh nenachádza v mriežke tak môžu nastať 4 prípady:

horná strana štvorca prechádza mriežkou, alebo dolná, alebo ľavá, alebo pravá.

Označíme si teda túto stranu kde sa nachádza – potom vieme, že $n*2+1$ políček príslušným smerom od tejto strany sú flaky, takže viem kde sú začiatky riadkov a tým pádom aj flaky.

Ak sa však ani jedna strana štvorca nenachádza v mriežke, tak vyfarbíme celú mriežku.

Zapísanie rohov príp. strán štvorcov nám trvá $O(r*s)$ času,

Potom prechádzame mriežku len po jednotlivých typoch rohov, pretože začiatky, prípadne konce riadkov sú na rôznych stranách, no a potom už len každý začiatok, alebo koniec riadku vieme prejsť ako v riešení s väčšou pamäťou.

Počet typov rohov, hrán a začiatkov a koncov riadkov je konečný počet cca 10, takže čas sa nám tiež nezvýši, ale pamäť sa nám zníži na $O(r*s)$.

Toto riešenie je tiež správne a vždy dá správny výsledok, lebo som rozobral všetky možné prípady aké môžu nastať pri pamäti $r*s$.