Opisi algoritama

IBT 2011. MASTERS LIGA - 5. kolo

Komentare i pitanja uputite na askurdija@gmail.com *ili na* frane.kurtovic@gmail.com.

Utovar (P1-1)

Za svaku dimenziju skladišta izračunat ćemo koliko zadanih kocaka stane s obzirom na tu dimenziju. U tu svrhu treba nam cjelobrojno dijeljenje. Primjerice, L div A broj je kocaka koji stane s obzirom na dimenziju L. Ukupno dakle stat će $(L \text{ div } A) \cdot (W \text{ div } A) \cdot (H \text{ div } A)$ kocaka, što još množimo s A^3 da dobijemo ukupan zauzeti volumen. Potom zauzeti volumen oduzimamo od ukupnog volumena (koji iznosi $L \cdot W \cdot H$) da bismo dobili traženi volumen slobodnog prostora.

Boćanje (P1-2, P2-1)

Za svaku boću izračunat ćemo njezinu udaljenost do bulina poznatom formulom za udaljenost dvaju točaka u koordinatnoj ravnini. Potom ćemo, za svaki tim, sortirati te udaljenosti od manjih prema većima. Pobjednika je sada lako odrediti: to je tim čija je prva udaljenost u sortiranom nizu manja. Određivanje broja bodova svodi se na brojanje udaljenosti pobjedničkog tima koje su manje od prve udaljensoti u sortiranom nizu gubitničkog tima.

Pijuni (P1-3, P2-2)

Za svaku dijagonalu (pripazimo, postoje dvije vrste dijagonala s obzirom na smjer) riješit ćemo problem zasebno. Idući po dotičnoj dijagonali pamtimo trenutni broj slobodnih polja (od posljednjeg crvenog polja). Ako je on veći ili jednak K, povećavamo broj načina za jedan.

Valja pripaziti na poseban slučaj $K = 1$, jer stavimo li jedan pijun na ploču, on će pripadati dvjema dijagonalama, pa u tom slučaju rješenje dobiveno gornjim algoritmom dijelimo s 2.

Tvrtka (P1-4)

Promatrajmo graf u kojem su vrhovi danih N studenata, a između studenta A i studenta B postoji usmjereni brid ako je student A izjavio da želi biti nadređeni studentu B. Ovaj graf ne može imati ciklusa, jer rješenje u tom slučaju ne postoji. Stoga je dobiveni graf DAG (eng. *directed uncyclic graph*, tj. usmjereni graf bez ciklusa).

Poznato je da je svaki DAG moguće topološki sortirati, tj. poredati mu vrhove na način da svi bridovi idu u istom smjeru (npr. slijeva na desno u načinjenom nizu vrhova). Načinimo li topološki sort u našem grafu, možemo primijetiti da tražena hijerarhija može biti upravo lanac koji tako dobivamo (ne zaboravite, lanac je ujedno i stablo), budući da će svaki brid tada ići od nadređenog ka podređenomu, tj. svi zahtjevi bit će ispunjeni.

Mrav (P2-3)

Mjesto da računamo vjerojatnost da će mrav barem jednom posjetiti određeni kamen, izračunat ćemo komplementarnu vjerojatnost da mrav *nikada* neće posjetiti određeni kamen, pa onda tu vjerojatnost oduzeti od 100%.

A tu pak vjerojatnost računamo dinamičkim programiranjem. Zanima nas, preciznije, $p(x, k)$ kao vjerojatnost da mrav u prvih k minuta nikada ne posjeti kamen x (naravno, u konačnici nas zanima $p(x,n)$). Što možemo reći o $p(x,k)$? Najprije, očito je $p(x,k) = p(-x,k)$ pa promatrajmo samo slučaj $x > 0$ (za $x = 0$, $p(x,k) = 0$ jer mrav svakako posjećuje nulti kamen barem na početku).

Da bismo za $x > 0$ i $k > 0$ pronašli $p(x,k)$, promatrajmo prvi korak mrava.

- Ako ide lijevo tj. na kamen -1 (vjerojatnost $1/3$), tražimo vjerojatnost da se mrav u sljedećih $k-1$ minuta nikada ne pomakne za $x+1$ koraka udesno - ona iznosi $p(x+1,k-1)$.
- Ako ide desno tj. na kamen 1 (vjerojatnost $1/3$), tražimo vjerojatnost da se mrav u sljedećih $k-1$ minuta nikada ne pomakne za $x-1$ koraka udesno - ona iznosi $p(x-1,k-1)$.
- Ako ostaje na kamenu 0 (vjerojatnost $1/3$), tražimo vjerojatnost da se mrav u sljedećih $k-1$ minuta nikada ne pomakne za x koraka udesno - ona iznosi $p(x,k-1)$.

Imamo dakle rekurzivnu relaciju

$$p(x,k) = \frac{1}{3}p(x+1,k-1) + \frac{1}{3}p(x-1,k-1) + \frac{1}{3}p(x,k-1)$$

pomoću koje dinamički računamo tražene vjerojatnosti.

Koordinatni (P1-5, P2-4)

Točka (x,y) pripada opisanom trokutu stranice k ako i samo ako je $x+y \leq k$. Stoga je za određenu točku bitan samo njezin $z = x+y$. Sažimanjem svih učitanih brojeva z i k možemo stoga svesti problem na sljedeći: potrebno je podržati operacije

- povećaj broj točaka na određenom z za 1;
- za dani k, izračunaj broj točaka na svim $z \leq k$;

pri čemu su svi brojevi veličine do 200000.

Ovaj problem rješavamo uz pomoć logaritamske strukture, koja podržava operacije povećavanja nekog broja u nizu te pronalaženja sume prvih k brojeva u nizu.

Charlie (P2-5)

Za razumijevanje ovog rješenja potrebno je poznavati množenje matrica te pojam inverza matrice.

Ako pojedinu traku promatramo kao jednostupčanu matricu, tada postoji jedinstvena matrica A koja, množeći tu matricu, kao rezultat daje jednostupčanu matricu koja predstavlja *sljedeću* traku. Kako konstruirati tu matricu? Nije teško zaključiti da u i-tom njezinom retku trebaju biti same nule, osim jedinica na mjestima sa kojih i-to polje nove trake zbraja paketiće.

Tako, ako je x matrica koja predstavlja prvu traku, onda matrica Ax predstavlja drugu, A^2x treću, i tako dalje - $A^{m-1}x$ je jednostupčana matrica koja predstavlja posljednju traku.

Ako je dakle $A^{m-1}x = y$, onda je $x = (A^{m-1})^{-1}y$. Matricu A generiramo kako je rečeno, matrica y zadana je u ulaznim podacima - stoga, da bismo pronašli početnu matricu x, trebamo izračunati najprije A^{m-1} - što rješavamo poznatim algoritmom brzog potenciranja - te inverz dobivene matrice, što rješavamo Gauss-Jordanovom eliminacijom.