Miðmisseriskeppni TÖL607G 2022

Lausnayfirlit

Atli Fannar Franklín

23. febrúar 2022

Aukaskilskröfur

Við munum nú fara í yfirlit á lausnum dæmanna á miðmisseriskeppninni. Hins vegar munum við ekki gefa neinn kóða, bara yfirlit. Það er vegna þess að hægt er að fá aukaskil fyrir að leysa dæmin eftir keppni, fyrir 20:00 á sunnudag. Leysa þarf a.m.k. 4 dæmi af 7 til að fá þau skil metin. Opnað verður hólf á Kattis fyrir þessi skil. Við athugum að það að leysa 4 þessarra dæma er þyngra verkefni en venjuleg vikuskil, en á móti kemur að þið fáið hér drög að lausnum.

Dæmið

Gefin eru x_1, x_2, y_1, y_2 og svo n punktar (x_i, y_i). Segja á hversu margir punktanna (x_i, y_i) eru inní ferhyrningnum með gagnstæða hornpunkta (x_1, y_1) og (x_2, y_2) og hliðar samsíða hnitakerfisásunum.

Lausn

Ljóst er að (x_i, y_i) er inní ferhyrningnum þá og því aðeins að $x_1 \leq x_i \leq x_2$ og $y_1 \leq y_i \leq y_2$. Athugum að dæmið bendir sérstaklega á að þetta séu veikar ójöfnur.

Dæmið

Gefnar eru blöðrur og hvað þarf mörg högg til að sprengja þær. Einnig eru gefnar blöðrur sem koma úr iðrum fyrri blaðra þegar sigrast er á þeim. Finnið heildarfjölda höggva sem þarf mátað við $10^9 + 7$.

Lausn

Höldum utan um fjölda höggva sem þarf í heildina fyrir hverja blöðrutegund jafn óðum. Nota má t.d. `map<string,long long>` til þess í C++. Þá má reikna heildarfjölda höggva fyrir nýja tegund með því að leggja saman niðurstöðuna fyrir hverja blöðru sem kemur úr iðrum hennar ásamt höggvafjölda sem er gefið fyrir þessa blöðru.

Inntakið er síðan leyst með því að leggja saman það sem er talið upp með því að nota gögnin að ofan. Mikilvægt er að máta við $10^9 + 7$ jafn óðum til að fá ekki overflow. Best er að nota `long long` eða aðra 64-bita gagnategund þó svo að tæknilega séð sé hægt að reikna þetta í 32-bitum án yfirflæðis. Það er bara óþarflega flókið vesen.

Dæmið

Gefin er níu stafa tala x (mögulega með núllum fremst). Breytið mest 3 tölustöfum í x svo 2 gangi sem oftast upp í x. Hvað gengur 2 í mesta lagi oft upp í x eftir breytingu?

Lausn

Hvað er hægt að velja 3 tölustafi í x á marga vegu? Það er $\binom{9}{3} = 84$ og ítra má yfir það með þrefaldri for-lykkju. Hvað má breyta hvern tölustaf á marga vegu? Ekki þarf að breyta neinum tölustaf, svo setja má gildið sem var. Því eru einfaldlega 10 valkostir fyrir hvern staf. Samtals eru þetta 84000 leiðir til að breyta x og má ítra yfir það allt með margfaldri for-lykkju.

Þá er einfaldlega að gá hvað 2 gengur oft upp í hverja af þessum niðurstöðum og taka max. Telja má hversu oft 2 gengur upp í n með því að deila n með 2 meðan hún er slétt og telja hversu oft það er gert. Einnig má spara sér skriftir í `C++` með því að nota bara `__builtin_ctz`.

Passa þarf að ef maður breytir x í 000000000 þarf að continue-a, því 0 er ekki gilt gildi.

Dæmið

Gefnar eru n hæðir af óvinum. Hver hæð hefur einhverja óvini sem standa í röð, hver með styrktarstig c_{ij} og umbótargildi t_{ij}. Hetjan byrjar með styrktarstig s. Aðeins má berjast við óvin fremst á hverri hæð. Ef $s > c_{ij}$ getur hetjan sigrað og s hækkar þá upp í t_{ij}. Hvert er hæsta styrktarstig sem hægt er að ná?

Hraðapæling

Hugmyndin er að berjast við óvini á einhverri hæð þar til fremsti er of sterkur því maður tapar aldrei á því að afgreiða óvin strax.

En sú aðstaða getur komist upp að berjast þarf við óvinina í mjög tiltekinni röð. Ef styrktarstigin eru $0, 1, 2, \ldots, K$, $s = 1$ og $t_{ij} = 1$ fyrir öll i, j er bara hægt að vinna ef óvinir eru sigraðir í nákvæmlega réttri röð.

Því getur verið of hægt að renna í gegnum hæðirnar í vaxandi röð aftur og aftur. Þurfum einhverja gagnagrind sem heldur utan um hvar er best að fara næst.

Lausn

Notum forgangsbiðröð sem setur minnsta gildið efst. Setjum í hana pörin (c_{i0}, i), þ.e. styrktarstig óvinsins sem er fremst á hverri hæð ásamt því hvaða hæð hann er á. Tökum ávallt efsta gildið úr biðröðinni. Ef við getum barist við óvininn gerum við það og setjum hann sem er fyrir aftan í biðröðina (ef það er einhver fyrir aftan). Ef við getum ekki barist er verkefni lokið því við komumst ekki lengra.

Dæmið

Velja á $t \geq 0$ þannig að $b^t + c/(t+1)$ sé sem minnst fyrir gefin b, c.

Lausn

Látum $f(t) = b^t + c/(t+1)$. Þá er $f''(t) = \ln(b)^2 b^t + 2c/(t+1)^3$. Þar sem $t \geq 0$ og $b > 1$ gefur þetta að $f''(t) \geq 0$, svo f er kúpt. Því til að finna minnsta gildi f getum við einfaldlega notað þriðjungaleit.

Dæmið

Halda þarf utan um safn af punktum (x, y) og afgreiða fjórar tegundir af beiðnum:

1. Bæta (x, y) í safnið.
2. Fjarlægja (x, y) úr safninu.
3. Fyrir gefið c, hvert er minnsta og stærsta y-gildi allra punkta í safninu með x-gildi jafnt c.
4. Fyrir gefið c, hvert er minnsta og stærsta x-gildi allra punkta í safninu með y-gildi jafnt c.

Athugasemdir

Athugum að við þurfum ekki að svara fyrirspurnum jafn óðum. Við getum lesið inn allt inntakið, unnið með það sem heild, og prentað síðan út svörin.

Ímyndum okkur að við búum til fall sem svarar beiðnum gerðinni 3 en ekki 4. Þá gætum við svarað beiðnum af gerðinni 4 með því að svissa á x og y gildum punktanna áður en fallið fær gögnin.

Einnig ef við erum með fall sem gefur stærsta y-gildið í hverri fyrirspurn getum við fengið minnsta gildið í hverri fyrirspurn með því að breyta (x, y) í $(x, -y)$ áður en fallið fær gögnin.

Því dugar að svara bara hvert stærsta y-gildið er í fyrirspurn 3 og keyra það fall 4 sinnum með örlitlum breytingum á inntaki.

Röðunarlausn

Söfnum saman öllum (x, y) sem eru í inntakinu. Röðum þeim í vaxandi röð eftir x-gildi og númerum þau síðan $0, 1, 2, \ldots, k-1$. Búum til biltré af stærð k þar sem hvert gildi er $-\infty$ í byrjun. Þegar við bætum við punkti (x, y) finnum við hvert númer þess er og táknum það með i. Setjum þá i-ta gildi biltrésins sem y. Eins ef við fjarlægjum (x, y) finnum við þetta i og setjum i-ta gildið sem $-\infty$. Passa þarf sértilfelli þegar fleiri en einum jöfnum punkti er bætt við í safnið í inntakinu.

Nú svörum við fyrirspurn fyrir gefið c. Ef enginn starfsmaður í inntakinu hefur $x = c$ er svarið `Enginn!`. Annars finnum við fremsta og aftasta i, j þannig að starfsmaður i og j hafi $x = c$ með helmingunarleit. Þá biðjum við einfaldlega um max á bilinu $[i, j]$ í biltrénu. Ef niðurstaðan er $-\infty$ er svarið `Enginn!` en annars skilum við svarinu.

Dæmið

Gefin eru n, b og n hlutmengi $B_1, \ldots, B_n$ í $\{1, 2, \ldots, b\}$ sem hvert hefur kostnað c_i. Þegar tvö sammengi eru sameinuð fæst sammengi þeirra nema við hendum út öllu sem kemur fyrir í báðum mengjunum. Fyrir mengi A, B er þetta yfirleitt táknað $A \triangle B$.
Finnið ódýrasta safn mengja $B_{i_1}, \ldots, B_{i_k}$ þannig að $B_{i_1} \triangle \ldots \triangle B_{i_k} = \{1, 2, \ldots, b\}$ (eða skilið að ekkert slíkt safn sé til).

Aðgerðin $\triangle$

Munum nota bitakenna til að tákna hlutmengi. Þurfum þá b bita, $b \leq 16$ svo `int` virkar (eða aðrar 32-bita heiltölur). Viljum þá að $A_1 \triangle \ldots \triangle A_k$ sé jafnt `111...1`, þ.e. `(1 << b) - 1`. En hvernig reiknum við $\triangle$? Viljum að kveikt sé á bitunum sem eru til staðar í A eða B en ekki báðum. Þetta er oft nytsamlegt í tölvunarfræði, nóg til þess að til sé sér skipun fyrir þetta. Hún heitir XOR og er táknum með `^` í flestum málum. Við reiknum þá $A \triangle B$ bara með `A ^ B`.

DP

Við viljum nú finna lágmarkskostnað þess að komast upp í `(1 << b) - 1`. Getum gert þetta með kvikri bestun. Setjum upp rakningavensl sem ættu að vera svipuð venslum sem þið hafið séð áður. Látum $f(A, i)$ vera ódýrustu leiðina til að fá mengið A með því að nota bara $B_1, \ldots, B_i$. Þá er

$$f(A, i) = \begin{cases} 0 \text{ ef } A = \varnothing \\ \infty \text{ ef } i < 0 \\ \min(f(A, i-1), f(A \triangle B_i, i-1) + c_i) \text{ annars} \end{cases}$$

Reikna má þá upp úr þessu með kvikri bestun á þessari rakningarformúlu. Svarið er þá `f((1 << b) - 1, n - 1)`. Athugum að í hægu máli eins og Python gæti þurft að beita smá auka brögðum til að gera þetta nógu hratt.