

PSA2: DN2

*To ni dovolj,
želim si še več*

Nina P.

Od Jožeta k trenutno najbolj znanemu vzgojitelju Leopoldu. Dela v ogromnem vrtcu, ki sprejme do 10^6 otrok. Pri tako velikem številu otrok ni časa za vsakega posebej, zato je edina aktivnost hoja v vrsti po poteh okoli vrtca, hkrati pa mora vladati strogi red, zato so otroci v vrsto razvrščeni glede na svojo vpisno številko.

Včasih monotonost kateremu od otrok začne presedati, zato zajoka. Leopold je kot blisk pri njem (na mestu i v vrsti) in mu za tolažbo podari x bonbončkov. Včasih to vzbudi zavist pri otroku, ki je v vrsti za jokajočim (na mestu $i + 1$), zato zajoka in Leopold mu da $x + 1$ bonbončkov (ker jih je dobil kasneje kot prvi, le x ni dovolj). Včasih se to nadaljuje pri otrocih na mestih $i + 2$, $i + 3 \dots, j$, ki po vrsti dobijo $x + 2$, $x + 3$, \dots , $x + (j - i)$ bonbončkov. To operacijo označimo kot $\text{posodobi}(i, j, x)$.

Ker otroci bonbone le kopičijo, jedo jih pa ne, Leopolda skrbi, da bi razlike v zbrani količini bonbonov spet pripeljale do joka, zato ga zanima, kako nevarni so posamezni deli vrste. Njegova mera za nevarnost na odseku $[i, j]$ temelji na razliki števila bonbonov dveh zaporednih otrok in je enaka

$$f(i, j) = \max_{i \leq k < j} |b_{k+1} - b_k|,$$

kjer je b_k število do danega trenutka zbranih bonbonov otroka na mestu k .

Naloga A (5 točk)

Izvedi posodobitve $\text{posodobi}(i, j, x)$, nato pa – ko se število bonbonov ne spreminja več – izračunaj nekatere vrednosti $f(i, j)$. Dodatna olajšava: vpisna številka i -tega otroka je kar i (šteti začnemo z 0).

Naloga B (5 točk)

Posodobitve $\text{posodobi}(i, j, x)$ in izračuni $f(i, j)$ se dogajajo v mešanem vrstnem redu, vpisna številka pa je 14-mestno število¹, npr. 20210325505007.

¹Natančna oblika ni važna, oponaša pa EMŠO: 1111mddssabc, kjer 1 pomeni leto, m mesec, d dan, s spol (500 ali 505), abc pa so tri števke.

Vhodni podatki

Vsaki od štirih podnalog (A1, A2, B1 in B2) pripadajo vhodni datoteki (s končnicama `.in` in `.vpisna`) in izhodna datoteka s končnico `.out`. V `.vpisna` datotekah so v naraščajočem vrstem redu našteje vpisne številke – vsaka v svoji vrstici. V `.in` datotekah so vrstice `p vpisna0 vpisna1 x` in `p vpisna0 vpisna1`, kot kažeta primera. Zahtevajo podisi posodobitev podatkov (`p`) bodisi odgovor (`o`).

Pri posodobitvah bo $\text{vpisna0} \leq \text{vpisna1}$. Pri poizvedbah velja stroga neenakost (sicer iščemo maksimum prazne množice).

Število otrok v posamezni podnalogi ne presega 10^6 , število posodobitev ne presega 10^5 , prav tako ne število poizvedb. Iskane vrednosti $f(i, j)$ ne presegajo 10^6 .

Naloga

V datoteki `resitev.h` vas že čaka razred `Resitev` z javno dostopno metodo `int obdelaj(const string & ukaz)`. Implementirajte jo. Kot lahko razberete iz `main.cpp`, je `ukaz` preprosto ena od vrstic iz vhodne datoteke. Njen izhod je pri posodobitvah poljuben, pri poizvedbah pa naj bo enak $f(i, j)$, kjer indeksa i in j pripadata vpisnima številka iz poizvedbe. Če želite, lahko razširite razred z ustreznimi polji in metodami.

Datoteke `main.cpp` ne spreminjajte, lahko pa jo uporabite za preverjanje rešitev. Za povsem pravilno se šteje rešitev, ki je dovolj hitra in daje pravilne odgovore.

Opazili boste, da se čas izvajanja določi ob zagonu datoteke `main.exe`, saj imamo različno hitre računalnike².

Končno verzijo kode oddajte na učilnici (samo izvirno kodo), ker jo bomo izvajalci predmeta še enkrat preverili.

²Na 7-letnem prenosniku se vsak od primerov izvede v približno 600 ms (z uradno rešitvijo), dovoljen čas pa je približno 1800 ms.

Primer

Poleg glavnih nalog sta tu še primera vhoda za obe težavnosti (A0 in B0):

A0.vpisna:	A0.in:	A0.out:
0	p 0 0 10	9
1	p 0 4 10	1
2	o 0 1	14
...	o 1 4	
6	o 1 5	
7		

B0.vpisna:	B0.in:	B0.out:
20210101505001	p 20210101505001 20210101505001 10	10
20210102505001	o 20210101505001 20210102505001	9
20210103505001	p 20210101505001 20210105505001 10	14
...	o 20210101505001 20210102505001	
20210107505001	o 20210102505001 20210106505001	
20210108505001		

Pri nalogi A0 je po prvi posodobitvi stanje bonbonov $b = [10, 0, 0, \dots, 0]$. Po drugi posodobitvi je stanje bonbonov $b = [20, 11, 12, 13, 14, 0, 0, 0]$. Razlike so torej na koncu $[9, 1, 1, 1, 14, 0, 0]$, zato so odgovori na poizvedbe $9 = \max\{9\}$, $1 = \max\{1, 1, 1\}$ in $14 = \max\{1, 1, 1, 14\}$.

Pri nalogi B0 se spreminja le 8. številka (od 1 do 8) vpisne številke, sicer pa sta posodobitvi enaki kot zgoraj, kar se tiče indeksov v vrsti. Zato je odgovor na prvo poizvedbo $10 = \max\{10\}$ (prvi otrok ima 10 bonbonov, ostali 0), odgovora na drugo in tretjo (ko že imamo končno stanje bonbonov) pa $9 = \max\{9\}$ in $14 = \max\{1, 1, 1, 14\}$.

Namig

Ko premislite, kako posodobitev posodobi(i, j, x) vpliva na razlike, se koda z vaj za intervalne poizvedbe kar ponuja za prilagoditev.