13. travnja 2021.

2021 Natjecanje iz informatike

OPISI ALGORITAMA

Agencija za odgoj i obrazovanje
Education and Teacher Training Agency

HRVATSKI SAVEZ INFORMATIČARA

Ministarstvo znanosti, obrazovanja i sporta

5.1. Zadatak: Asteroidi Autor: Nikola Dmitrović

U trima različitim varijablama pratimo koliko ima asteroida koje vrste. Znamo da će jedan od tri broja sigurno biti nula. Uspoređivanje brojeva koji nisu nula lako odredimo većeg i povećamo odgovarajuću varijablu za jedan. Postupak ponovimo za sve asteroide.

Kategorija: ad hoc

5.2. Zadatak: Notepad Autor: Nikola Dmitrović

Idejno relativno jednostavan zadatak, implementacijski nešto zahtjevniji.

Na početku sve riječi spremimo u niz *notepad*.

Pitanje #1 i #2: Za svaki ćemo redak unaprijed odrediti broj riječi u njemu. To ćemo postići simulacijom dodavanja riječi iz *notepada* u redak sve dok riječ koju pokušavamo dodati stane u taj redak.

Pitanje #3: Algoritam koji smo osmislili za prvo pitanje ponavljat ćemo mijenjajući duljinu retka. Početna duljina retka treba biti jednaka broju riječi koje imamo.

Pitanje #4: Iz niza *notepad* obrišemo zadane riječi i ponovimo algoritam iz prvog pitanja.

Kategorija: ad hoc

5.3. Zadatak: Jambolino Autor: Josip Klepec

Zadatak se mogao riješiti na dva načina.

Jedan od načina je isprobavanje svih mogućih kombinacija te zatim za svaku od njih izračunati broj bodova te naposljetku uzeti onu kombinaciju koja je rezultirala s najviše bodova.

Ovaj način je veoma koristan jer se mnogo problema s njima može riješiti. Čitatelja upućujem da pogleda naredbu next_permutation() za jezik c++ ili naredbu itertools.permutations() za programski jezik python iz paketa itertools (potrebno je napraviti import itertools)

Drugi način je rješavati zadatak pohlepno. Primijetimo da ako možemo složiti i jamb i poker, uvijek će se više isplatiti složiti poker. Slično ako imamo 2 jamba i 2 pokera, za jamb ćemo uzeti ono bacanje gdje je veća vrijednost kocke. Isto vrijedi i za tris…

Sada možemo napraviti za svaku od kombinacija (tris, poker, jamb) niz veličine 6 ili 6 varijabli. To nam predstavlja redom koliko tris-a postoji s vrijednosti 1, s vrijednosti 2, …, s vrijednosti 6. Isto za poker i jamb.

Sada idemo redom:

Prvo za jamb uzmemo najbolju kombinaciju (ako postoji) te u našem pomoćnom nizu maknemo tu kombinaciju iz pokera i trisa. Zatim isto radimo za poker te maknemo odabranu najbolju kombinaciju iz trisa.

Primijetimo da bismo ovim načinom mogli riješiti zadatak i da je bilo više bacanja (a ne samo 4).

Na posljetku još pogledamo može li koje bacanje biti skala. Jedno bacanje ne može u istom trenutku biti skala i još neka druga kombinacija pa se oko toga ne moramo brinuti.

Kategorija: ad hoc

6.1. Zadatak: Radionice Autor:

Učenike obrađujemo prema broju bodova na županijskom natjecanju počevši od onog s najviše. Za svakog učenika prolazimo po radionicama redom kao što je zadano u njihovim prioritetima i dodjeljujemo mu prvu radionicu u kojoj još ima mjesta. Za to je potrebno u dodatnom nizu pamtiti kojem broju učenika je dodijeljena pojedina radionica.

Kategorija: ad hoc

6.2. Zadatak: Obitelji Autor: Pavel Kliska

Za dvije obitelji možemo reći da su istog oblika ako imaju isti broj dječaka i djevojčica. Obitelji dvije djevojčice ili dva dječaka **i**, **j** su istoga oblika ako vrijedi **Xi**=**Xj** i **Yi**=**Yj**, a obitelji dječaka **i** i djevojčice **j** su istoga oblika ako vrijedi **Xi**+1=**Xj** i **Yi**=**Yj**+1.

Za odgovor na prvo pitanje možemo pomoću dvije ugniježđene for petlje proći po svim parovima dječaka i djevojčica. Ukoliko neki par dječaka i djevojčice ima isti oblik obitelji moguće je da su brat i sestra pa treba ispisati „DA”, ako ne postoji takav par treba ispisati „NE”.

Odgovor na drugo pitanje možemo odrediti tako da za oblik obitelji svakoga djeteta pronađemo svu ostalu djecu koja imaju isti oblik obitelji. Ako među tom djecom ima barem onoliko dječaka i barem onoliko djevojčica koliko ima u tom obliku obitelji treba ispisati „DA”, a ako ne postoje takva djeca treba ispisati „NE”.

U trećem pitanju bilo je potrebno za svaki oblik obitelji odrediti koliko najmanje obitelji tog oblika postoji i zbrojiti te brojeve. Ako u jednom obliku obitelji ima **M** dječaka i **F** djevojčica te postoji **m** dječaka i **f** djevojčica s tim oblikom obitelji, minimalan broj obitelji tog oblika je: max(ceil(**m**/**M**), ceil(**f**/**F**)). Ako je u toj formuli **M** ili **F** nula, promatra se samo vrijednost ceil(**f**/**F**) odnosno ceil(**m**/**M**).

Za četvrto pitanje je bilo dovoljno pri izračunu minimalnog broja obitelji za svaki oblik obitelji u rješenje dodati broj dječaka i djevojčica u toj obitelji.

Kategorija: ad hoc

6.3. Zadatak: Jamb Autor: Josip Klepec

Pogledaj opis zadatka Jambolino. S tim da u ovom zadatku moramo još staviti full ispred trisa. Dakle jamb -> poker -> full -> tris.

Kategorija: ad hoc

7.1. Zadatak: Editor Autor: Stjepan Požgaj

Vidi opis 5.2.

Kategorija: ad hoc

7.2. Zadatak: Lice Autor: Gabrijel Jambrošić

Zadatak se može riješiti na puno različitih načina. Jedan od načina je da stvorimo novu matricu u koju ćemo spremiti očekivani izgled lica na temelju uzoraka koje otkrijemo prolazeći kroz učitanu matricu.

Tako npr. možemo iterirati po matrici po redovima i stupcima dok ne dođemo do prvog popunjenog polja. Zatim iteriramo po tom istom redu dok ne dođemo do praznog polja. Neka smo na taj način iterirali kroz *a* zauzetih polja. Kada bi matrica sadržavala lice, onda bi taj prvi segment od a zauzetih polja trebao biti gornji rub lijevog oka. Zatim možemo u našoj novoj matrici zauzeti kvadrat veličine *a* x *a* polja počevši od iste pozicije kao u učitanoj matrici. Također zauzmemo i desno oko, simetričan kvadrat s obzirom na okomitu središnju os matrice.

Na sličan način možemo zauzeti očekivana polja za nos i usta. Za nos možemo tražiti prvo polje u sredini matrice koje počinjemo tražiti ispod očiju te odredimo visinu segmenta po kojem se protežu zauzeta polja. Zatim u novoj matrici zauzmemo trokutasti oblik s očekivanim oblikom nosa.

Za usta tražimo prvo zauzeto polje ispod nosa, ali ovaj puta iteriramo u horizontalnom smjeru da nađemo duljinu usta.

Nakon što smo prošli sve komponente i zauzeli očekivana polja u novoj matrici, ostaje još provjeriti poklapaju li se dobivena i naša matrica. Ako se ne poklapaju, onda postoji dio u dobivenoj matrici koji ne odgovara očekivanom obliku lica. Treba još pripaziti da je rub matrice prazan te da postoji razmak između komponenti.

Vremenska složenost: $O(N^2)$

Kategorija: ad hoc

7.3. Zadatak: Vlakovi — Autor: Josip Klepec

Zadatak zapravo idejno nije veoma kompliciran. Najteži dio u zadatku je napraviti funkciju koja će biti u stanju za dva vlaka i njihova vremena polaska biti u stanju odrediti krše li oni pravila zadana u zadatku. Nakon što to imamo zadatak riješavamo na sljedeći način:

- Vlakove obrađujemo redom
- Cijelo vrijeme pamtimo trenutno vrijeme polaska zadnjeg vlaka
- Kada obrađujemo novi vlak pokušavamo njega poslati u isto vrijeme kao i prethodni vlak (moramo provjeriti za sve vlakove koji su već krenuli je li to moguće). Ako jest vrijeme polaska tog vlaka je određeno, a inače pomičemo vrijeme polaska za 5 minuta pa opet provjeravamo i tako redom dok ne pronađemo vrijeme kada taj vlak može krenuti.

Još moramo vidjeti kako efikasno odrediti za 2 vlaka poštuju li sva pravila.

Možemo imati funkciju koja prima početnu i završnu postaju prvog vlaka, početnu i završnu postaju drugog vlaka te koliko minuta kasnije drugi vlak kreće od prvog.

Implementaciju funkcije pogledajte u izvornom kodu.

Neke stvari koje nam mogu olakšati implementaciju su:

- Umjesto da stalno radimo sa intervalima od 5 minuta, vrijeme možemo prikazati prirodnim brojem tako da je vlakovima između stanica potrebna 1 minuta, a kasnije sve pomnožimo sa 5
- U funkciji pomaknemo početnu postaju ondje gdje će se on nalaziti u onom trenutku kada kreće drugi vlak (ako je prvi vlak u tom trenutku već završio s vožnjom, onda nema problema). Inače, problem je nešto nalik problemu "sijeku li se ova dva intervala"

Kategorija: ad hoc

8.1. Zadatak: Grijalica

Autor: Marin Kišić

Za 20 bodova dovoljno je bilo isprobati sve X i Y.

U ostalim bodovima treba primijetiti da ako mi fiksiramo X, onda bi Y trebao biti T/X. Ako T/X nije u intervalu [C, D], samo preskočimo taj par, a inače ga normalno obradimo kao u prvom podzadatku. Ako bismo naivno isprobali sve moguće X, to nas dovodi do 52 boda.

Za sve bodove trebalo je zaključiti da će i X biti djelitelj od T i da je onda dovoljno isprobati sve X u intervalu [1, $\sqrt{T}$]U [A, B]. Iz ovog vidmo da će vremenska složenost biti O($\sqrt{T}$).

Kategorija: ad hoc

8.2. Zadatak: Smajlić

Autor: Gabrijel Jambrošić

Vidi zadatak 7.2 Lice.

Kategorija: ad hoc

8.3. Zadatak: Bazen

Autor: Vedran Kurdija

U prvom podzadatku bilo je dovoljno za svaku osobu proći po svakoj minuti u kojoj se ona nalazi u bazenu te za svaku od tih minuta proći po svim drugim osobama i provjeriti je li njih barem M u bazenu, tj. treba li tu minutu pridodati rješenju. Složenost je O(N^2*O).

U drugom podzadatku ovaj je pristup valjalo malo ubrzati. Napravimo niz u kojeg ćemo za svaku minutu spremiti koliko je ljudi tada bilo na bazenu. Za svaku osobu prođimo po svim minutama u kojima je bila na bazenu te pripadajuća mjesta u nizu povećajmo za jedan. Nakon što smo to učinili za sve osobe, ponovno prolazimo po svim osobama te za svaku osobu prolazimo po svim minutama u kojima je bila na bazenu, ali sada za provjeru treba li minutu pridodati rješenju ne moramo proći po svim drugim ljudima, jer nam je broj ljudi koji su tada u bazenu već spremljen u niz te je potrebno samo provjeriti je li taj broj barem M. Ako jest, pridodajemo tu minutu rješenju. Složenost je O(N*O).

Za sve bodove postupak je trebalo dodatno ubrzati. Za svaku osobu pohranimo u pomoćni niz tri vrste događaja: dolazak čovjeka na bazen u minuti D, kraj plaćenog perioda za tog čovjeka u minuti D+P-1 te odlazak osobe s bazena u minuti D+O-1. Sortirat ćemo ove događaje po minutama i prolaziti po njima redom, s time da ako više događaja dijeli istu minutu, prvo prolazimo dolaske, potom krajeve plaćenih perioda, a potom odlaske. Tijekom prolaska, u jednoj pomoćnoj varijabli pamtit ćemo koliko je ljudi trenutno na bazenu, a u varijabli *minuta* u koliko je minuta dosad na bazenu bilo barem M ljudi. Kada naiđemo na događaj dolaska čovjeka, povećavamo broj ljudi na bazenu. Kada naiđemo na kraj plaćenog perioda, u pomoćni niz za tog čovjeka spremamo trenutnu vrijednost varijable *minuta*. Kada naiđemo na odlazak čovjeka, od trenutne vrijednosti varijable *minuta* oduzimamo staru vrijednost koju smo spremili na kraju njegovog plaćenog perioda te time dobivamo broj minuta samo za njegov neplaćeni period. Naposljetku, pri odlasku čovjeka smanjujemo broj ljudi na bazenu. Varijablu *minuta* ažuriramo konstantno, prilikom svakog dolaska i odlaska osobe s bazena. Složenost prolaska je O(N), ali složenost sortiranja je O(N*log(N)), što je onda i konačna složenost rješenja.

Kategorija: ad hoc