

## Zadatak: Fontana

Za uspješno rješavanje zadatka ključno je odrediti sljedeća tri podatka:

1. broj dužina svakog mlaza vode
2. kut između dvije dužine u prvom mlazu vode
3. promjenu tog kuta između dva susjedna mlaza

Kada odredimo te podatke, nije problem  $n$  puta nacrtati po jedan mlaz, s odgovarajućim brojem dužina, te promijeniti kut među dužinama za crtanje sljedećeg mlaza.

Broj dužina svakog mlaza vode (označimo ga s  $m$ ) dobivamo sljedećom formulom:

$$m = \frac{(a+d-1)}{d}, \text{ zaokruženo na manji prirodan broj}$$

Kada imamo taj rezultat, kut između dužina u prvom mlazu (analiziranjem kutova) dobivamo kao

$$kut = \frac{180 - \frac{k}{2}}{m-1}$$

Sada imamo dovoljno podataka za nacrtati prvi, srednji i posljednji mlaz.

Razlika tih kutova između svaka dva susjedna mlaza je  $kutr = \frac{2\text{ kut}}{n-1}$ , pa možemo lako nacrtati sve mlazove vode.

Za rješavanje testnih primjera vrijednih 10% bodova, dovoljno je odrediti kut između dvije dužine prvog i posljednjeg mlaza, a taj kut iznosi  $180 - \frac{k}{2}$ .

Za rješavanje testnih primjera vrijednih 20% bodova (slučaj  $n = 3$ ), dovoljno je odrediti broj dužina u mlazovima vode te kut između dužina u prvom i posljednjem mlazu,  $kut = \frac{180 - \frac{k}{2}}{m-1}$ .

Za rješavanje testnih primjera vrijednih 20% bodova (slučaj kada svaki mlaz vode ima 2 dužine), dovoljno je odrediti kut između dužina kod prvog i posljednjeg mlaza ( $180 - \frac{k}{2}$ ) te razliku tog kuta među susjednim mlazovima ( $kutr = \frac{2\text{ kut}}{n-1}$ ).

## Zadatak: Loto

Za osvajanje 60% bodova na ovom zadatku, dovoljno je bilo crtati niz kružnica, tj. po jednu kružnicu u svakom stupcu. Pritom je bilo potrebno paziti na redoslijed ispune - to je moguće učiniti, primjerice, provjeravanjem ostatka rednog broja kružnice pri dijeljenju s 3. Ovisno o tome je li ostatak 0, 1, ili 2, krug se ispunjava crvenom, zelenom ili plavom bojom.

Za pravilno ucrtavanje zvijezdi, potrebno je provjeravati je li broj koji kružnica predstavlja prost, tako da se nekom petljom provjeri ima li taj broj djelitelja, tj. je li ostatak pri dijeljenju tog broja s brojevima manjim od njega jednak 0.

Za osvajanje preostalih 40% bodova, potrebno je bilo implementirati povećanje broja krugova kroz stupce i pravilan redoslijed crtanja. Najjednostavniji način je početi svaki stupac crtati iz njegovog središta, te zatim provjeriti crtaju li se krugovi unutar njega prema gore ili prema dolje,

te se ovisno o tome pomaknuti za odgovarajuću udaljenost prema gore ili prema dolje i pravilno usmjeriti.

## Zadatak: Igra

S obzirom na to da su potezi zadani u obliku  $[x\ y]$ , gdje može vrijediti i  $x > y$  i  $y > x$ , korisno je napraviti zamjenu ta dva elementa ako vrijedi  $x > y$ , tako da je uvijek prvi element poteza manji od drugog.

Za kvadrat s oznakom  $X$ , vrijedi da kvadrati iznad, ispod, lijevo i desno od njega imaju oznake redom  $X - m$ ,  $X + m$ ,  $X - 1$ ,  $X + 1$  ako ti kvadrati postoje ( $X$  se ne nalazi u prvom/zadnjem stupcu/retku).

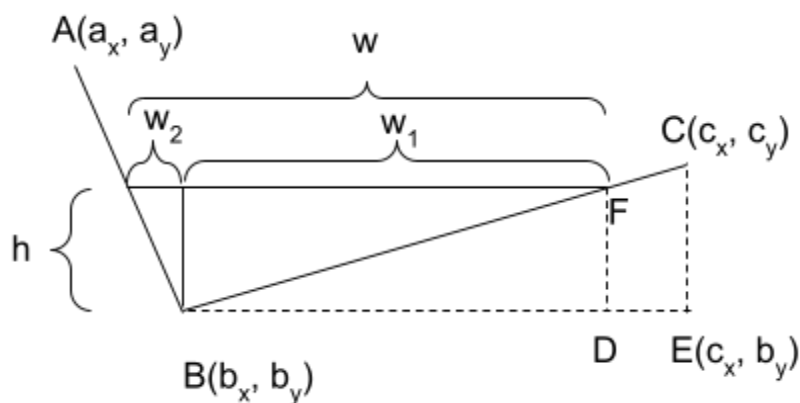
Pri rješavanju obrađujemo redom poteze. Za svaki potez provjeravamo je li valjan (jesu li kvadrati koje spajamo susjedni, postoje li oznake kvadrata i je li potez već izvršen) a zatim provjeravamo je li neki kvadrat zatvoren. Da kvadrat  $X$  bude zatvoren, moraju biti izvršeni potezi  $[X-m\ X]$ ,  $[X\ X+1]$ ,  $[X-m+1\ X+1]$ ,  $[X-m\ X-m+1]$ . Potezom  $[X\ Y]$  moguće je potencijalno zatvoriti kvadrate  $X$ ,  $Y$ ,  $Y-1$ ,  $Y+1$ , stoga je potrebno provjeriti jesu li izvršeni prethodno navedeni potezi za ta četiri kvadrata.

## Zadatak: Kiša

Za prvu skupinu testnih primjera bilo je dovoljno nacrtati reljef pomoću naredbe SETPOS i ispuniti ga naredbom FILL.

Za sve ostale test primjere trebalo je nacrtati trokute vode u dolinama. Neka tri točke:

$A(a_x, a_y)$ ,  $B(b_x, b_y)$ ,  $C(c_x, c_y)$  određuju jednu takvu dolinu.



Površina vode u ovoj dolini bit će  $k(c_x - a_x)$ , no iz toga nije odmah jasno kako će izgledati trokut. Pretpostavimo da ipak nekako znamo nacrtati trokut i probajmo ga okarakterizirati. Za izračun površine trokuta čini se najjednostavnije uzeti  $h$ , visinu iz točke  $B$  na površinu vode, i  $w$ , duljinu površine vode. Mora vrijediti  $\frac{1}{2}hw = k(c_x - a_x)$ , no to je sustav sa dvije nepoznanice  $h$  i  $w$ , koji ima više rješenja. Srećom, vrijednosti  $h$  i  $w$  nisu nezavisne.

U jednoj skupini test primjera kut na dnu doline uvijek je bio  $90^\circ$  i simetrala doline je sjekla  $x$  os pod pravim kutem. Tada je trokut vode zapravo bio polovica kvadrata i veza između  $w$  i  $h$  je bila  $w = 2h$ .

U težoj skupini testnih primjera trebalo je uočiti da visina iz točke  $B$  na površinu vode dijeli trokut na dva sukladna pravokutna trokuta. Nadopunimo jedan od tih manjih trokuta do pravokutnika s točkom  $D$ .  $\triangle BDF$  je sukladan promatranom trokutu i sličan  $\triangle BEC$  (oba su pravokutna i dijele kut u točki  $B$ ). Koristeći sličnost trokuta dobivamo omjer kateta trokuta  $w_1$ :  $|BE| = h$ :  $|CE|$ . Iz tog

omjera imamo  $w_1 = h \frac{|BE|}{|CE|} = h \frac{c_x - b_x}{c_y - b_y}$ . Kako su u ovoj skupini testnih primjera dva trokuta

sukladna imamo  $w = 2w_1 = 2h \frac{c_x - b_x}{c_y - b_y}$ .

U svim preostalim testnim primjerima nije bilo ograničenja na kut na simetralu kuta na dnu doline pa trokutni nisu bili sukladni, no i dalje je bilo moguće analogno odrediti ovisnost  $w_2$  o  $h$

koja je:  $w_2 = h \frac{b_x - a_x}{a_y - b_y}$ . Tada je  $w = w_1 + w_2 = h \left( \frac{c_x - b_x}{c_y - b_y} + \frac{b_x - a_x}{a_y - b_y} \right)$ .

U svim ovim razmatranjima smo za vezu između  $w$  i  $h$  dobili izraz oblika  $w = Kh$  za neki realan

$K$ . Uvrštavanjem u formulu za površinu dobivamo  $\frac{1}{2}Kh^2 = k(c_x - a_x)$ , odnosno  $h = \sqrt{\frac{2k(c_x - a_x)}{K}}$ .

Iz  $h$  se mogu odrediti  $w_1$  i  $w_2$  pa samo ostaje iskoristiti naredbe SETPOS i FILL za crtanje i popunjavanje trokuta.

## Zadatak: Parking

Osnovna ideja rješenja je promatrati sve događaje kronološkim redoslijedom, pri čemu uz zadane događaje treba promatrati i one kada postoji mogućnost naplate kazne.

Situacije kada postoji mogućnost naplate kazne su:

- 6 minuta od dolaska automobila na parking
- *trajanje*\*60 + 6 minuta od kupovine parkirne karte
- *trajanje*\*60 + 1 minuta od isteka trenutne karte, u slučaju produživanja.

Zato se u svakoj od navedenih situacija na odgovarajuće mjesto u listi dodaje novi događaj tipa 4, s oznakom vremena i registracije na koju se odnosi.

Međutim, događaji tipa 4 su samo potencijalni trenuci kada će biti izdana kazna, odnosno nije sigurno da će u svakom od tih trenutaka kazna i biti izdana (brzo vidimo da imamo višak tih događaja).

Na primjer, ako je karta kupljena unutar 5 minuta, kazna neće biti izdana 6 minuta nakon dolaska, iako i tada postoji događaj tipa 4.

Preciznije, kako bi kazna zaista i bila izdana, nužno je još (a i dovoljno) i sljedeće:

- automobil u tom trenutku zaista je na parkiralištu, odnosno u međuvremenu nije otišao
- u tom trenutku ne postoji ispravna karta
- taj trenutak nije unutar 5 minuta od zadnjeg dolaska

Svaki od ovih slučajeva rješavamo pamćenjem informacija:

- o automobilima koji su trenutno na parkiralištu (u službenom rješenju lista *auti*)
- o isteku parkirnih karti za svaki od automobila za koji je karta ikad kupljena (u službenom rješenju lista *karte*)
- o isteku 5 minuta od posljednjeg dolaska (u službenom rješenju lista *istek*)

Preostaje samo implementirati ovo rješenje prolaskom po listi događaja i obradom događaja na gore opisani način (jedan od načina je micanje prvog elementa liste dok god lista nije prazna). Treba paziti i na pohranjivanje i uklanjanje odgovarajućih informacija u listama s automobilima na parkiralištu, istekom karti i istekom 5 minuta od dolaska.

Jedna moguća implementacija je u službenom rješenju.

U zadatku je bilo ponuđeno i nekoliko podzadataka koji su nosili manji broj bodova. U nastavku je rješenje za svaki od tih podzadataka:

1. (za 10% bodova) – U ovom podzadatku dovoljno je provjeriti je li uopće kupljena karta za vozilo. Ako nije, kazna se izdaje 6 minuta nakon dolaska. Ako je karta kupljena, pitanje je kada, zbrojeno sa svim produžavanjima, ta karta ističe (treba uzeti u obzir i da karta može biti kupljena prije dolaska automobila). S obzirom na uvjete zadatka, zapravo je garantirano da će automobilu biti izdana točno jedna kazna.
2. (za 20% bodova) – Ovaj podzadatak je proširenje prethodnog. Potrebno je, između svakog dolaska i odlaska automobila, primijeniti postupak opisan u 1. podzadatku (primijetite kako registracija ovdje nema značajnu ulogu s obzirom na zadane uvjete).
3. (za 10% bodova) – Dovoljno je provjeriti koji su automobili stigli, nisu otišli unutar 5 minuta, a za koje nije uopće kupljena karta.
4. (za 20% bodova) – Ovaj podzadatak navodi do potpunog rješenja – ideja za rješenje je u osnovi ista, samo što je implementacija rješenja znatno jednostavnija.

## Zadatak: Karta

Pristup rješavanju zadatka je rekurzivan, a postupak je sljedeći:

1. za svaki grad treba provjeriti s kojim je sve gradovima povezan
2. grad treba povezati crtom odgovarajuće duljine sa svakim susjednim gradom (osim grada iz kojega je taj grad nacrtan) u odgovarajućem redoslijedu i odgovarajućem rasporedu
3. u svakom susjednom gradu treba ponoviti korake 1.-3.

Pritom treba posebno provjeriti slučaj kada se nalazimo u gradu 1, jer tada ne postoji grad iz kojega je on nacrtan (drugim riječima, potrebno je bez iznimke crtati sve gradove s kojima je povezan). Moguća su dva pristupa tom problemu: prvi je crtati grad 1 i njemu povezane gradove izvan rekurzije (te u svakome od njemu povezanih gradova pozivati rekurzivnu proceduru), a drugi je modificirati kutove rotacije u rekurziji kada se nalazimo u gradu 1 (ta implementacija se nalazi u službenom rješenju).

U koraku 2., za svaki je grad potrebno proći po zadanoj listi te izdvojiti sve gradove s kojima je povezan u drugu listu, koju zatim treba sortirati.

Primijetite da ovako zadana rekurzija završava u svakom gradu koji je povezan samo s jednim gradom (onim iz kojeg je taj grad nacrtan). Naime, u tim gradovima se koraci 2. i 3. praktički izostavljaju.

Implementacija ove rekurzije (u kojoj se i rješava problem duljine crta) nalazi se u službenom rješenju.

Test primjeri vrijedni ukupno 50% bodova (u tri slučaja navedena u bodovanju), mogu se jednostavno riješiti bez primjene rekurzije.