

2 Diena 2 Uždavinys

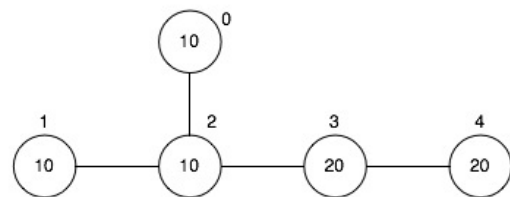
Eismo spūstis

Nors Kanada yra didžiulė šalis, kurioje yra daugybė neapgyvendintų vietų, tačiau dauguma gyventojų susitelkę ties pietine siena. 1962 metais nutiestas 7 821 kilometro ilgio transkanadinis greitkelis jungia šią labiausiai gyvenamą pietinę juostą pradėdant rytuose esančiu St. Johno miesteliu ir baigiant Viktorijos miestu vakaruose.

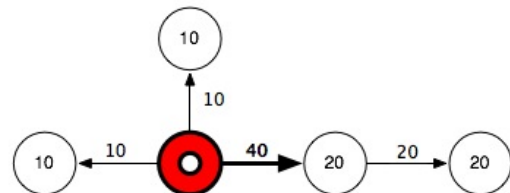


Kanadiečiai mėgsta ledo ritulį. Po žaidynių tūkstančiai gerbėjų sėda į mašinas ir vyksta namo sudarydami eismo spūstis keliuose. Turtingas verslininkas nori pastatyti naują areną jo įsigytai ledo ritulio komandai. Padėkite jam parinkti vietą arenos statybai minimizuodami eismo spūstis po žaidynių.

Miestai sujungti kelių tinklu. Visi keliai yra dvikrypčiai, kiekvieną porą miestų jungia lygiai vienas maršrutas. Maršrutą iš miesto c_0 į miestą c_k nusako miestų seka c_0, \dots, c_k – tokia, kad kiekvienam i yra kelias iš miesto c_{i-1} į miestą c_i . Naujoji arena turi būti statoma viename iš miestų, jį pavadinsime *žaidynių mieste*. Po ledo ritulio turnyro visi gerbėjai vyksta iš žaidynių miesto į savo miestus, žinoma, išskyrus tuos, kurie gyvena tame mieste. Kiekvieno kelio spūsties mastas proporcingas tuo keliu vykstančių ledo ritulio gerbėjų skaičiui.

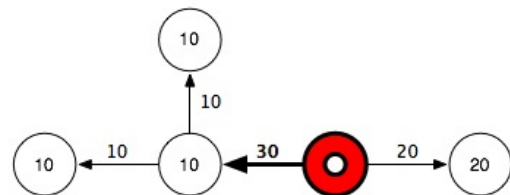


Parinkite miestą arenos statybai taip, kad spūsties mastas labiausiai užkimštame kelyje būtų kiek galima mažesnis. Jei yra keletas vienodai tinkamų miestų arenos statybai, pasirinkite bet kurį vieną.



Parašykite procedūrą **LocateCentre**(N, P, S, D).

Sveikasis teigiamas skaičius N reiškia miestų skaičių. Miestai sunumeruoti nuo 0 iki $N - 1$. P yra N teigiamų sveikųjų skaičių masyvas: $P[i]$ reiškia i -jame mieste gyvenančių ledo ritulio gerbėjų skaičių. Bendras ledo ritulio gerbėjų skaičius visuose miestuose neviršys 2 000 000 000. Masyvai S ir D , kiekviename kurių yra po $N - 1$ skaičių, skirti keliams aprašyti. Kiekvienas i nurodo kelią, jungiantį miestus, kurių numeriai $S[i]$ ir $D[i]$. Procedūros rezultatas turi būti sveikasis skaičius, reiškiantis miesto, kuriame bus statoma arena, numerį.



Pavyzdys.

Viršuje dešinėje pateikta penkių miestų tinklo diagrama, kurioje 0-ajame, 1-ajame ir 2-ajame miestuose gyvenančių ledo ritulio gerbėjų skaičius lygus 10, o 3-ajame ir 4-ajame miestuose – 20. Vidurinioji diagrama vaizduoja spūstį, jei naujoji arena būtų pastatyta 2-ajame mieste:

2 Diena 2 Uždavinys

Eismo spūstis

didžiausia spūstis tada būtų lygi 40 (pavaizduota paryškinta rodykle). Apatinė diagrama vaizduoja spūstį, jei naujoji arena būtų pastatyta 3-iajame mieste: didžiausia spūstis būtų lygi 30 (pavaizduota paryškinta rodykle). Taigi naujos arenos statybai 3-iasis miestas yra tinkamesnis negu 2-asis. Šio pavyzdžio duomenys pateikti `grader.in.3a` faile.

1 Uždutis [25 taškai]. Duomenys tokie, kad visi miestai išsidėstę vienoje linijoje iš Rytų į Vakarų ir visi keliai eina šia linija be jokių išsišakojimų.

Tiksliau sakant, kiekvienam i galioja: $0 \leq i \leq N - 2$, $S[i] = i$, ir $D[i] = i + 1$.

Miestų skaičius neviršija 1 000.

2 Uždutis [25 taškai]. Galioja tos pačios prielaidos kaip ir 1 uždutyje, tik miestų skaičius neviršija 1 000 000.

3 Uždutis [25 taškai]. 1 uždutyje nurodytos prielaidos nebegalioja.

Miestų skaičius neviršija 1 000.

4 Uždutis [25 taškai]. 1 uždutyje nurodytos prielaidos nebegalioja.

Miestų skaičius neviršija 1 000 000.

Realizacija.

- naudokite *RunC programavimo ir derinimo aplinką*
- Dalyvio katalogas: `/home/loi2010-contestant/traffic/` (prototipas: *traffic.zip*)
- Reikia realizuoti: `traffic.c` arba `traffic.cpp` arba `traffic.pas`
- Dalyvio sąsaja: `traffic.h` arba `traffic.pas`
- Vertinimo programos sąsaja: nėra
- Pavyzdinė vertinimo programa: `grader.c` arba `grader.cpp` arba `grader.pas`
- Pavyzdinės vertinimo programos duomenų failas: `grader.in.1`, `grader.in.2`
Pastaba: pirmoje duomenų failo eilutėje įrašytas skaičius N . Tolesnėse N eilučių įrašyti $P[i]$ kur i kinta nuo 0 iki $N - 1$. Tolesnėse $N - 1$ eilučių įrašytos poros $S[i]$ $D[i]$ kur i kinta nuo 0 iki $N - 2$.
- Pavyzdinės vertinimo programos pradinių duomenų failo rezultatas: `grader.expect.1`, `grader.expect.2` ir t.t.
- Kompiliavimas ir vykdymas (komandinė eilutė): `runc grader.c` arba `runc grader.cpp` arba `runc grader.pas`
- Kompiliavimas ir vykdymas (gedit įskiepis): *Control-R*, redaguojant uždavinį sprendžiančią procedūrą.
- Pateikimas (komandinė eilutė): `submit grader.c` arba `submit grader.cpp` arba `submit grader.pas`
- Pateikimas (gedit įskiepis): *Control-J*, redaguojant bet kurį procedūros arba vertinimo programos failą.