

Xedef

Xedef pasiuntinių kompanija teikia siuntinių gabenimo oru paslaugas tarp kelių miestų. Kai kurie šių miestų yra Xedef *centrai*, kuriuose yra įsikūrusios ypatingos siuntinių apdorojimo įstaigos. Kiekvienas Xedef lėktuvas skraido pirmyn ir atgal tarp dviejų konkrečių miestų, gabendamas siuntinius, kai reikia.

Siunčiant siuntinį iš vieno miesto į kitą, siuntinys yra transportuojamas žingsnių seka. Kiekvienu žingsniu siuntinys gabenamas iš vieno miesto į kitą, kuriuos jungia tiesioginis skrydis. Be to, sekoje turi būti bent vienas Xedef centras.

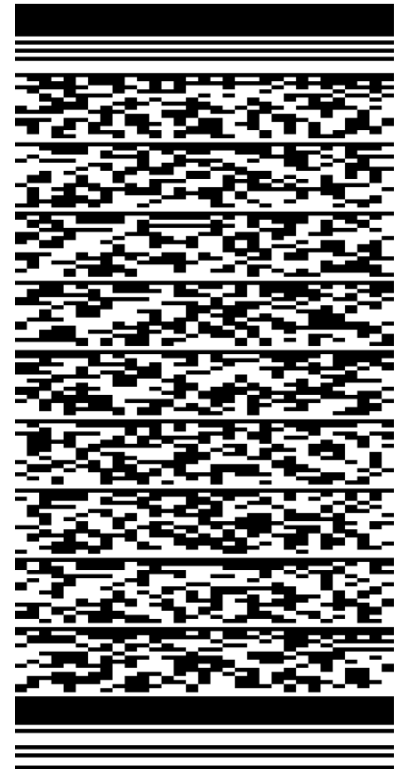
Xedef nori etiketėje, klijuojamoje ant kiekvieno siuntinio, užkoduoti trumpiausią žingsnių seką nuo kiekvieno miesto iki kiekvieno centro ilgius. (Trumpiausios žingsnių sekos nuo centro iki jo paties ilgis yra nulis.) Akivaizdu, kad reikia šią informaciją pateikti glaudžiai.

Jums reikia realizuoti dvi procedūras, **encode(N,H,P,A,B)** ir **decode(N,H)**. N yra miestų kiekis, o H – centrų kiekis. Miestai yra sunumeruoti nuo 0 iki $N-1$, o centrai yra miestai, turintys numerius nuo 0 iki $H-1$. Galioja ribojimai $N \leq 1000$ ir $H \leq 36$. P yra miestų porų, kuriuos jungia tiesioginis skrydis, kiekis. Visos miestų poros yra skirtingos. A ir B yra masyvai, kurių dydis yra P ; pirma sujungtų miestų pora yra $(A[0], B[0])$, antra pora yra $(A[1], B[1])$ ir t. t.

encode turi sudaryti bitų seką, pagal kurią **decode** gali nustatyti žingsnių kiekį nuo kiekvieno miesto iki kiekvieno centro. **encode** perduos bitų seką vertinimo serveriui seka **encode_bit(b)** iškvietimų, kur b yra 0 arba 1. **decode** gaus bitų seką iš vertinimo serverio kviesdama procedūrą **decode_bit**. i -asis **decode_bit** iškvietimas grąžins b reikšmę iš i -ojo **encode_bit(b)** iškvietimo. Jūs turite užtikrinti, kad **decode** iškviestų **decode_bit** tiek pat arba mažiau kartų, nei **encode** iškvietė **encode_bit(b)**.

Iškodavusi žingsnių kiekius, **decode** turi iškvieti **hops(h,c,d)** kiekvienam centrui h ir kiekvienam miestui c (įskaitant ir centrus, t. y., ir kai $c = h$), pateikdama mažiausią žingsnių kiekį d , reikalingą siuntinio pergabenimui tarp h ir c . Taigi, **hops(h,c,d)** turi būti iškviesta $N \cdot H$ kartų. Iškvietimų tvarka nėra svarbi. Visada įmanoma pergabenti siuntinį tarp kiekvieno centro ir kiekvieno miesto.

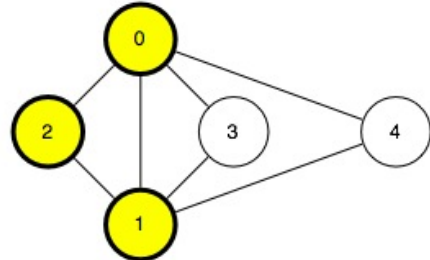
*Pastaba: **encode** ir **decode** turi bendrauti tik per pateiktą sąsają. Bendri kintamieji, failų ir tinklo prieiga yra draudžiama. C ir C++ kalbose galite aprašyti pastovius kintamuosius su **static tam**, kad jie liktų pastovūs, tačiau nebūtų bendri. Pascal kalboje galite aprašyti pastovius kintamuosius **implementation dalyje**.*



Xedef

Pavyzdys.

Pažvelkime į diagramą dešinėje. Joje matome penkis miestus ($N = 5$), kuriuos jungia septyni lėktuvai ($P = 7$). Miestai 0, 1 ir 2 yra centrai ($H = 3$). Siuntinio gabenimui tarp centro 0 ir miesto 3 pakanka vieno žingsnio, tačiau gabenimui tarp centro 2 ir miesto 3 reikia 2 žingsnių. Šio pavyzdžio duomenys yra faile `grader.in.1`.



Pateiktoje lentelėje yra visos d reikšmės, kurias **decode** turi perduoti kviesdama **hops(h,c,d)**:

d		City c				
		0	1	2	3	4
Hub h	0	0	1	1	1	1
	1	1	0	1	1	1
	2	1	1	0	2	2

1 Uždutis [25 taškai]. `encode` turi iškvieti `encode_bit(b)` ne daugiau nei 16 000 000 kartų.

2 Uždutis [25 taškai]. `encode` turi iškvieti `encode_bit(b)` ne daugiau nei 360 000 kartų.

3 Uždutis [25 taškai]. `encode` turi iškvieti `encode_bit(b)` ne daugiau nei 80 000 kartų.

4 Uždutis [25 taškai]. `encode` turi iškvieti `encode_bit(b)` ne daugiau nei 70 000 kartų.

Realizacija.

- naudokite RunC programavimo ir derinimo aplinką
- Dalyvio katalogas: `/home/ioi2010-contestant/saveit/` (prototipas: `saveit.zip`)
- Reikia realizuoti:
 - `encoder.c` arba `encoder.cpp` arba `encoder.pas`
 - `decoder.c` arba `decoder.cpp` arba `decoder.pas`
- Dalyvio sąsaja:
 - `encoder.h` arba `encoder.pas`
 - `decoder.h` arba `decoder.pas`
- Vertinimo programos sąsaja: `grader.h` arba `graderlib.pas`
- Pavyzdinė vertintinimo programa: `grader.c` arba `grader.cpp` arba `grader.pas` ir `graderlib.pas`

2 Diena 4 Uždavinys

Xedef

- Pavyzdinės vertintinimo programos duomenų failas: `grader.in.1`, `grader.in.2` ir t. t.
Pastaba: Pirmoje failo eilutėje yra skaičiai N P H . Tolesnėse P eilučių yra miestų poros $A[0]B[0]$, $A[1]B[1]$ ir t. t. Tolesnėse $H \cdot N$ eilučių yra žingsnių kiekis tarp kiekvieno centro ir kiekvieno miesto (įskaitant jį patį ir visus kitus centrus); t. y. žingsnių kiekis tarp centro i ir miesto j yra $(i \cdot N + j + 1)$ -oje iš šių eilučių.
- Pavyzdinės vertinimo programos pradinių duomenų failo rezultatas:
 - Jei procedūros realizuotos teisingai 1 užduočiai, programa išves OK 1
 - Jei procedūros realizuotos teisingai 2 užduočiai, programa išves OK 2
 - Jei procedūros realizuotos teisingai 3 užduočiai, programa išves OK 3
 - Jei procedūros realizuotos teisingai 4 užduočiai, programa išves OK 4
- Kompiliavimas ir vykdymas (komandinė eilutė): `runc grader.c` arba `runc grader.cpp` arba `runc grader.pas`
- Kompiliavimas ir vykdymas (gedit įskiepis): *Control-R*, redaguojant bet kurį procedūros failą.
- 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.