

D. Laser Strike

Ülesande nimi	Laser Strike
Ajapiirang	3 sekundit
Mälupiirang	1 gigabait

Olivia ja Birgit hiljuti avastasid uue lauamängu, millest on saanud nende lemmik: Laser Strike. Selles mängus töötavad kaks mängijat koos, et eemaldada N nuppu laualt. Mängus on kaks faasi. Konks on selles, et Birgitil pole täielikku informatsiooni mängu kohta. Et mängu võita, peavad Olivia ja Birgit töötama koos, aga samas suhtlema nii vähe kui võimalik.

Laual on N unikaalset nuppu, nummerdatud 0 kuni $N - 1$. Mõlemad mängijad näevad neid nuppe. Samuti on $N - 1$ ühendust nupupaaride vahel nõnda, et on võimalik igast nupust jõuda igasse teise minnes mööda ühendusi. Teisisõnu, need ühendused moodustavad puu. **Ainult Olivia näeb neid ühendusi, Birgit neid ei tea.**

Mängu esimeses faasis valib Olivia järjekorra $\ell_0, \ell_1, \dots, \ell_{N-2}$, milles nuppe eemaldada, kuni vaid üks on alles. Seda järjekorda hoitakse Birgiti eest saladuses. Kui ta suudab seda jäljendada, nad võidavad mängu. Nuppude eemaldamine peab rahuldama järgmist reeglit: iga kord, kui nupp eemaldatakse, see peab olema ühendatud täpselt ühe allesoleva nuppuga. Teisisõnu, see peab olema leht puus, mida moodustavad allesolevad nupud. (Pärast $N - 1$ nupu eemaldamist eemaldatakse viimane nupp automaatselt ja mängijad võidavad.) Olivia peab valima järjekorra, mis vastab sellele reeglile.

Olivia samuti kirjutab Birgitile sõnumi binaarsõnena. Olivia saab valida, kui pikk sõnum on, aga mida lühem see on, seda rohkem punkte nad saavad.

Pärast seda algab mängu teine faas. Mängu eesmärgiks on Birgitil laualt eemaldada $N - 1$ nuppu järjekorras $\ell_0, \ell_1, \dots, \ell_{N-2}$. Ta teeb $N - 1$ käiku. Enne käiku i ütleb Olivia Birgitile täisarvupaari a , b järgmiste omadustega:

- $a < b$;
- on endiselt otseühendusega nupupaar, numbritega a ja b ; ning
- kas a või b on õige nupp ℓ_i , mida peaks eemaldama sellel käigul.

Pane tähele, et Olivia jaoks on ühendus (a, b) üheselt määratud praeguse puu lehe ℓ_i järgi.

Birgit seejärel eemaldab laualt kas a või b . Kui see on õige nupp ehk ℓ_i , nad jätkavad mängu. Muul juhul nad kaotavad mängu.

Sinu ülesandeks on implementeerida nii Olivia kui Birgiti strateegiad, et nad võidaksid mängu.

Sinu programmi hinnatakse Olivia mängu esimeses faasis kirjutatud sõnumi pikkuse põhjal.

Implementatsioon

Tegemist on kaheetapilise interaktiivse ülesandega ehk sinu programmi käivitatakse kaks korda. Esimene kord, kui seda käivitatakse, implementeerib see Olivia strateegiat mängu esimeses faasis. Pärast seda, implementeerib programm Birgiti strateegiat mängu teises faasis.

Esimesel real on kaks täisarvu P ja N , kus P on kas 1 või 2 (esimene või teine faas) ning N on nuppude arv.

Järgnev sisend sõltub faasist:

1. faas: Olivia

Pärast esimest rida (kirjeldatud eespool), on sisendi järgmisel $N - 1$ real puu kirjeldus. Igal real on kaks arvu, a ja b ($0 \leq a < b \leq N - 1$), mis tähendab, et nuppude a ja b vahel on ühendus.

Sinu programm peaks esmalt väljastama binaarsõne, millel on kõige enam 1 000 märki, iga neist 0 või 1, Olivia kirjutatud sõnum.

Seejärel peaks programm väljastama $N - 1$ täisarvu $\ell_0, \ell_2, \dots, \ell_{N-2}$ eraldi ridadel, mis tähistavad, et mis järjekorras Olivia tahab eemaldada puu lehti. Järjekord peab olema selline, et kui nuppe eemaldatakse puust ühekaupa selles järjekorras, siis eemaldatud nupp on alati leht ehk puu peab alati sidusaks jääma.

2. faas: Birgit

Pärast esimest rida (kirjeldatud eespool), on sisendi järgmisel real binaarsõne (Olivia sõnum) esimesest faasist.

Pärast seda on $N - 1$ vooru interatsiooni, üks iga Birgiti käigu jaoks.

Käigul i peaks su programm sisse lugema kaks arvu a ja b ($0 \leq a < b \leq N - 1$). Üks nendest nuppudest on leht ℓ_i Olivia järjekorras ning teine nupp on ainus teine nupp ühendatud nupuga ℓ_i . Seejärel peaks su programm väljastama ℓ_i , mis tähendab, et Birgit eemaldab selle lehe. Kui su programm ei väljasta õiget lehte ℓ_i , siis tüdrukud kaotavad mängu ja sinu esitus saab hinnangu Wrong Answer.

Detailid

Kui programmi käitusaegade *summa* ületab ajalimiiti, saab su esitus hinnangu Time Limit Exceeded.

Jälgi, et tühjendaksid standardväljundi puhvrit pärast igat sõitu või muidu su programm võib saada hinnangu Time Limit Exceeded. Pythonis toimub see automaatselt, kui kasutad `input()`, et ridu lugeda. C++-s `cout << endl;` koos uue rea väljastamisega samuti tühjendab puhvrit; kui kasutad `printf`i, kasuta `fflush(stdout)`.

Tühja sõne lugemine võib olla keeruline. Antud mallid käsitlevad seda õigesti.

Piirangud ja hindamine

- $N = 1\,000$.
- $0 \leq a < b \leq N - 1$ iga ühenduse jaoks.

Sinu lahendust testitakse hulgal testigruppidel, iga neist on väärt mingi arvu punkte. Igas testigrupis on hulk teste. Et saada testigrupi eest punkte, pead läbima kõik gruppi kuuluvad testid.

Grupp	Maksimum skoor	Piirangud
1	8	Puu on täht. See tähendab, et kõik tipud peale ühe on lehed.
2	9	Puu on ahel. See tähendab, et kõikidel tippudel peale kahe lehe on kaks naabertippu.
3	21	Puu on täht, millest väljuvad ahelad. See tähendab, et igal tipul on üks või kaks naabertippu peale ühe, millel on rohkem naabertippe.
4	36	Kaugus kahe tipu vahel on enamalt 10.
5	26	Lisapiirangud puuduvad.

Iga testigrupi eest, mida su programm õigesti lahendab, saad punkte vastavalt järgmisele valemile:

$$\text{skoor} = S_g \cdot (1 - 0.3 \cdot \log_{10} \max(K, 1)),$$

kus S_g on testigrupi maksimumpunktide arv ning K on pikim sõnumipikkus, mida Olivia vajab ühegi testi jaoks testigrupis. **Sinu skoor iga testigrupi jaoks ümardatakse lähima täisarvuni.**

Järgmine tabel näitab punktide arvu, mõne K väärtuse puhul, mida su programm saab, kui see lahendab kõik testigrupid selle K -ga. Selleks, et saada 100 punkti, peab su lahendus lahendama igat testi nõnda, et $K \leq 1$.

K	1	5	10	50	100	500	1000
Punkte	100	79	70	49	39	20	11

Testimistööriist

Et hõlpsustada oma lahenduse testimist, on antud lihtne tööriist, mida saab alla laadida. Vaata “attachments” kattise ülesandelehe alumises otsas. Tööriist pole kohustuslik ning sul on lubatud seda muuta. Kattise ametlik hindaja on testimistööriistast erinev.

Tööriista kasutamiseks, loo sisendfail, näiteks “sample1.in, mis algab arvuga N , millele järgneb $N - 1$ rida, mis kirjeldavad puud, samas vormistuses, nagu on esimeses faasis. Näiteks järgnev näide:

```
7
0 1
1 2
2 3
0 4
0 6
1 5
```

Pythoni programmide jaoks, nagu `solution.py` (tavaliselt käivitatakse `pypy3 solution.py`):

```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

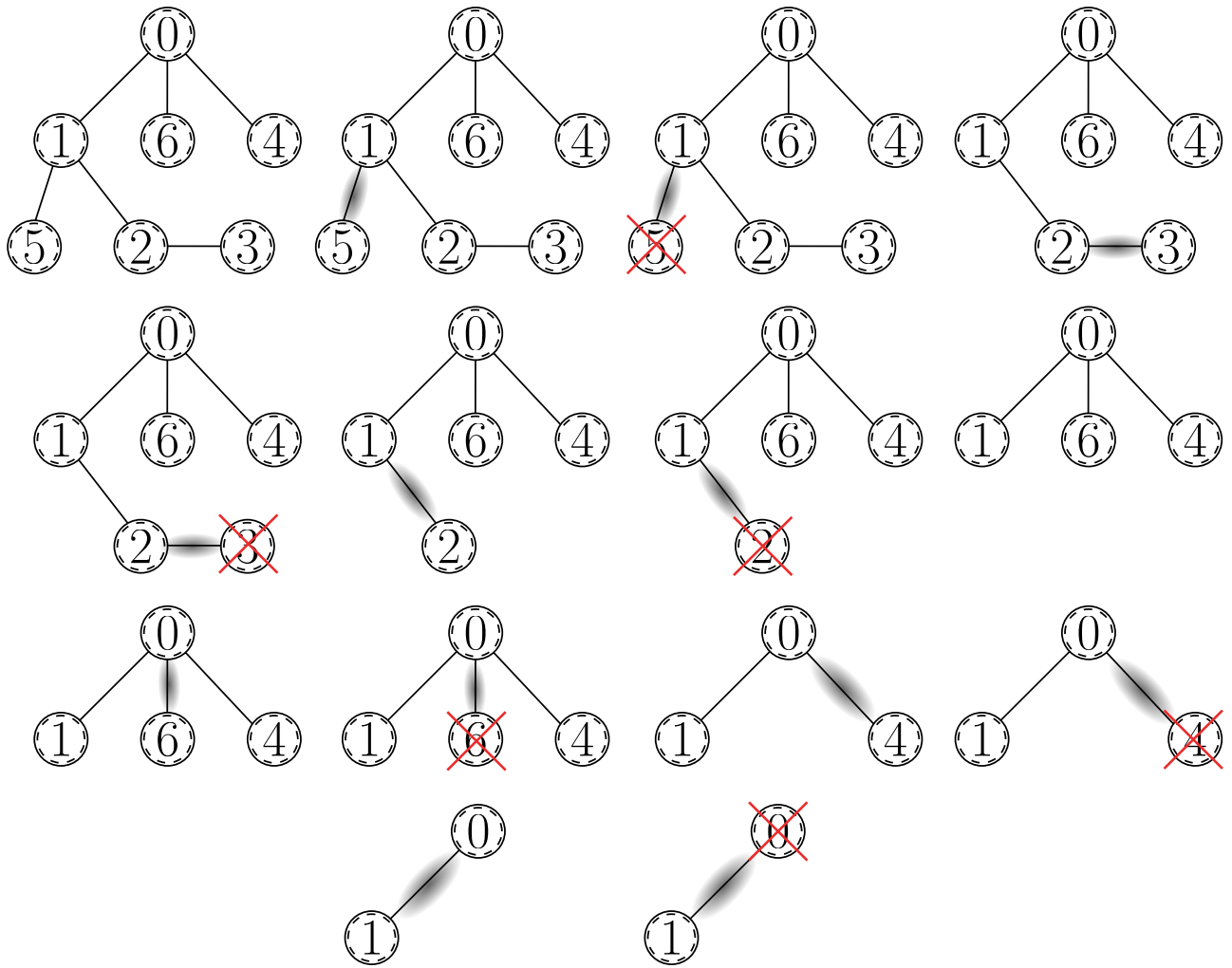
C++ programmide jaoks, kõige pealt kompileeri (näiteks kasutades `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) ja siis käivita:

```
python3 testing_tool.py ./solution.out < sample1.in
```

Näide

Pane tähele järgmises näites lihtsuse mõttes $N = 7$ ning pole seega korrektne test. Sinu programm ei pea lahendama seda juhtu. Kõikides hindaja testides $N = 1\,000$.

Selles näites saab Olivia järgmise puu. Esimeses faasis loeb Olivia sisse selle puu ning valib binaarsõne “0110”, mida Birgitile saata ning valib järjekorra $[\ell_0, \ell_1, \dots, \ell_{N-2}] = [5, 3, 2, 6, 4, 0]$, milles nuppe puust tuleks eemaldada. Teises faasis saab Birgit sõne “0110”, mida saadeti esimeses faasis. Ta seejärel saab paari $(1, 5)$ ning otsustab eemaldada tipu 5, mis on tõepoolest õige leht. Järgmiseks käiguks saab ta paari $(2, 3)$ ning eemaldab lehe 3 ja nii edasi. Järgnevad pildid näitavad interaktsioone:



hindaja väljund	sinu väljund
1 7	
0 1	
1 2	
2 3	
0 4	
0 6	
1 5	
	0110
	5
	3
	2
	6
	4
	0

hindaja väljund	sinu väljund
2 7	
0110	
1 5	
	5
2 3	
	3
1 2	
	2
0 6	
	6
0 4	
	4
0 1	
	0