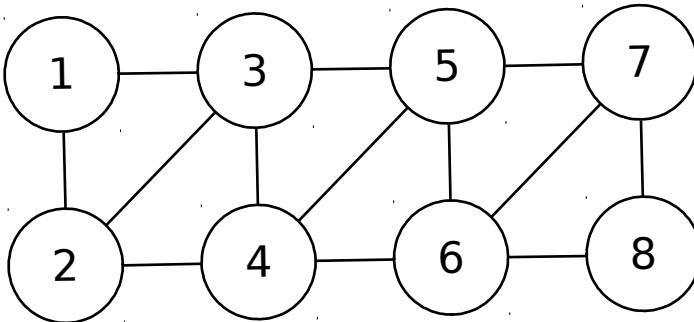


Pavle je odlučio da i ove godine posećuje niške kafane. Od svog vodiča Stošića saznao je da Niš ima oblik pravougaonika sa dva bulevara i da je svaki bulevar dugačak N kilometara. Na svakom od ovih bulevara nalazi se $N+1$ kafana, dakle ukupno $2N+2$ kafana. One su označene brojevima i povezane sokacima kao što je prikazano na slici ($N=3$ u ovom slučaju). Niš ima jedan ulaz (kod kafane broj 1) i jedan izlaz (kod kafane broj $2N+2$).

Svi sokaci u Nišu su jednosmerni i kada se Pavle šeta on mora da poštuje smer svakog sokaka. Sokaci su orijentisani **tako da nema ciklusa**. Takođe, ulaz u Niš je jedino mesto **od kojeg su svi sokaci izlazeći**, a izlaz iz Niša je jedino mesto **od kojeg su svi sokaci ulazeći**. Za sve druge kafane važi: **there postoji i ulazeći i izlazeći sokak**.



Kažemo da sokak ima **smer 1** ako ide od kafane sa manjim brojem ka kafani sa većim brojem, inače kažemo da ima **smer 0**. Pavle zna da dva sokaka kod ulaza u Niš i izlaza u Niš imaju smer 1 (pošto ulaz u Niš ima redni broj 1 a izlaz ima broj $2N+2$), ali ne zna ništa o smerovima drugih kafana i želi da ih sazna da bi mogao da isplanira svoj izlazak.

Za sve ljubitelje matematike i informatike (kao Pavle), pored hotela Nais postoji kompjuter sa specijalnim programom. Taj kompjuter odgovara na pitanja sledećeg oblika: za datu listu sokaka, gde su sokaci predstavljeni rednim brojevima kafana koje povezuju, program vraća XOR njihovih smerova. *Podsetnik, XOR dva binarna broja je 0 ako su oni jednaki, 1 ako su različiti. Ako ima više od dva operanda, prvo se XOR-uju prva dva pa se rezultat XOR-uje sa narednim i tako redom.* **Ako se programu da lista od samo jednog sokaka, program vraća njegov smer.**

Pavle želi da sa malim brojem pitanja otkrije smerove svih sokaka.

Komisija je kopirala program sa Naisa u sistem za testiranje. Pomozite Pavletu tako što ćete napisati funkciju `run`, koja se kompajlira zajedno sa programom komisije i koja će komunicirati sa njim postavljajući pitanja na gore opisan način i koja će ispravno pronaći smerove svih sokaka u Nišu.

Detalji implementacije

Vaša funkcija `run` mora imati sledeću deklaraciju:

```
void run(int n);
```

Ona se zove tačno jednom i kao argument dobija prirodan broj N - dužinu Niša (ukupan broj kafana je $2N+2$).

Za komunikaciju sa komisijskim programom možete pozivati sledeće dve funkcije:

```
bool get_xor(const std::vector<std::pair<int, int>>& edges);
```

```
void state_direction(const std::pair<int, int>& edge, bool direction);
```

Rezultat poziva funkcije `get_xor` je XOR svih smerova koji su navedeni u vektoru `edges`. Svaki sokak je predstavljen kao par rednih brojeva kafana koje spaja (redosled kafana unutar para nije bitan). Složenost funkcije `get_xor` (i vremenska i memorijska) je linearna po broju sokaka u vektoru.

Vaša funkcija mora da pozove `state_direction` za svaki sokak (ponovo predstavljen na isti način) sa otkrivenim smerom, koji se navodi u parametru `direction`. Ako u bilo kom

trenutku vaš program pozove neku od ovih funkcija sa nevalidnim parametrima ili funkciju `state_direction` sa pogrešnim smerom, dobićete 0 (slovima: nula) poena na tom test primeru. Takođe ćete dobiti nula (dekadno: 0) poena za test primer ukoliko postoje sokaci za koje niste pozvali `state_direction`.

Na sisten za testiranje šalžete jedan fajl **park.cpp**. U njemu, pored funkcije `run`, možete imati bilo koje pomoćne funkcije, promenljive, klase, struct-ove i slično, ali vaš fajl ne sme da sadrži funkciju `main` i mora da uključi heder `park.h` sa preprocesorskom instrukcijom `#include "park.h"`.

Ograničenja

$$1 \leq N \leq 100000$$

Bodovanje

Vaše rešenje će dobiti broj poena različit od nule, ako za svaki test primer uspešno izvrši zadatak unutar vremenskog ograničenja i broj poziva funkcije `get_xor` nije veći od $3N$. Broj poena se računa na sledeći način (Q je broj poziva funkcije `get_xor` u nekom test primeru; $\lceil x \rceil$ - najmanji ceo broj koji je $\geq x$):

15 poena : $Q \leq 3N$ važi za sve primere, ali $Q > 2N$ na bar jednom primeru.

45 poena: $Q \leq 2N$ važi za sve primere, ali $Q > \lceil \frac{3}{2}N \rceil$ na bar jednom primeru.

70 poena: $Q \leq \lceil \frac{3}{2}N \rceil$ važi za sve primere, ali $Q > \lceil \frac{16}{11}N \rceil$ na bar jednom primeru.

100 poena: $Q \leq \lceil \frac{16}{11}N \rceil$ za sve primere.

Lokalno testiranje

Dat vam je fajl `Lgrader.cpp`, koji možete kompajlirati zajedno sa svojim programom da biste ga testirali. Kada se on pokrene pitaće vas za N i nakon toga za smerove svih sokaka (0 ili 1). Nakon toga se štampa sva komunikacija. Možete menjati ovaj fajl po potrebi.

Testiranje na sistemu i adaptivni testovi

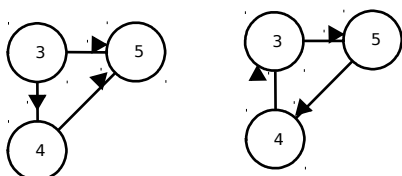
Na pojedinim test primerima (vama nepoznatim) komisijski program se adaptira postavljenim pitanjima, što znači da smerovi svih sokaka nisu unapred fiksirani. Ako testirate svoj program na sistemu za testiranje, nemate ovu opciju. Format ulaza na sistemu za testiranje je sledeći: Jedan prirodan broj N , nakon kojeg sledi $4N+1$ nula i jedinica - smerovi sokaka jedan za drugim (1-2, 1-3, 2-3, 2-4, ...).

Primer komunikacije

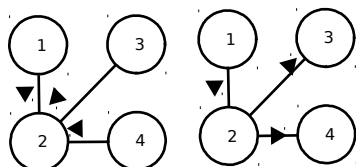
№	Akcije funkcije run	Akcije i odgovori komisijskog programa
1.		run(2)
2.	state_direction({1,2}, 1)	
3.	state_direction({1,3}, 1)	
4.	state_direction({4,6}, 1)	
5.	state_direction({5,6}, 1)	
6.	get_xor({{3,4}, {3,5}, {4,5}})	1
7.	get_xor({{3,5}})	1
8.	state_direction({3,4}, 1)	
9.	state_direction({3,5}, 1)	
10.	state_direction({4,5}, 1)	
11.	get_xor({{2,3}, {2,4}})	0
12.	state_direction({2,3}, 1)	
13.	state_direction({2,4}, 1)	
14.	The function terminates.	

Objašnjenje primera

- Komisijski program zove run sa $N=2$.
- 5. Run navodi smerove od ulaza u Niš i izlaza iz Niša.
- 7. Run pita za dva XOR-a i dobija odgovor 1 za oba. Posmatrajmo moguće slučajeve za smerove sokaka:



- 10. Druga opcija sadrži ciklus što nije moguće, pa run navodi smerove iz prve opcije (prve dve sličice).
- Run pita za XOR dva sokaka i dobija odgovor 0. Dakle, njihovi smerovi su isti:



- 13. U prvom slučaju svi sokaci susedni kafani 2 ulaze u nju, što nije dozvoljeno (osim za izlaz iz Niša odnosno kafanu $2N+2$), pa run navodi smerove iz druge opcije.
- Run se završava. Postavljena su 3 pitanja.