

Tapşırıq 1. Monopoliya oyunu

Deni monopoliya oyunu oynamaqdan çox həzz alır. Ancaq bir oyunun müddəti çox uzundur - 5 saatdan 7 saata qədər. Beləliklə, Deni klassik qaydaları dəyişdirmək barədə düşünməyə başlayır. Monopoliyada olan oyunçular adətən bir xanadan başqa bir xanaya bir istiqamətdə hərəkət edir və bir müddət sonra başlanğıca qayıdırlar, yenidən başlayırlar və s. Yeni versiyada hərəkət yenə bir xanadan başqa bir xanaya olacaq, lakin indiki versiyada növbəti gediş üçün bir neçə variant ola bilər. Deni xanalar arasında elə istiqamətlənmiş əlaqələr tapmaq istəyir ki, oyunçu necə hərəkət edirsə etsin (təbii ki, qaydalara əməl etməklə) heç vaxt əvvəl olduğu yerə birdə qayıtmınsın. Bu şəkildə oyunun müddəti daha qısa olacaq.

O, artıq yeni lövhəni düzəltməyə başlayıb – xanaların sayını N seçib (xanalar 1-dən N -ə qədər nömrələnir) və M sayda əlaqələr siyahısı tərtib edib (hər bir əlaqənin bir istiqaməti var və heç bir xananı özü ilə birləşdirən əlaqə yoxdur). Əgər i xanası j xanasına bağlıdırsa, onda əks istiqamətdə, yəni j xanasından i xanasına birbaşa əlaqə yoxdur, həmçinin i xanasından j xanasına başqa birbaşa əlaqə yoxdur. Deni yeni lövhənin hazır olduğunu düşündü, amma birdən gördü ki, istədiyi şərt (əlaqələrdən istifadə edərək, xanadan xanaya keçərkən heç vaxt daha əvvəl olduğu xanaya gəlməməlisən) onun tərtib etdiyi əlaqələr siyahısında ödənmir. O, əvvəlcə bəzi birbaşa əlaqələri silməyi düşündü, lakin bu, siyahının yenidən yazılması ilə nəticələənəcək və bu, həqiqətən uzun ola bilər. Buna görə Deni bəzi istiqamətləndirilmiş əlaqələrin istiqamətini dəyişdirmək qərarına gəldi.

Tapşırıq

Siz müntəzəm olaraq Deni ilə monopoliya oynayırınsız. Buna görə də ona **monopoly** proqramını yazmaqla kömək etmək istəyirsiniz, hansı ki, bu proqram ona istədiyi şərtin ödənməsi üçün hansı əlaqələrin istiqamətini tərsinə dəyişməli olduğunu deməlidir. Proqramda münşiflər heyətinin proqramı ilə kompayl olunacaq *find_reverse* funksiyası olmalıdır.

İmplementasiya detalları

find_reverse funksiyasının prototipi aşağıdakı kimi olmalıdır:

```
std::string find_reverse (int N, int M, int connections[][2]);
```

O, münşiflər heyətinin proqramı tərəfindən üç parametrlə yalnız bir dəfə çağırılır: N – yeni lövhədəki xanaların sayı, M – Deninin siyahısındakı istiqamətlənmiş əlaqələrin sayı və hər biri iki x və y – istiqamətlənmiş əlaqə üçün başlanğıc və son xanaları saxlayan M sayda sətirdən ibarət *connections* massivi. Bu funksiya M uzunluğunda ikili (binary) sətir qaytarmalıdır – *connections* massivinın sırasına uyğun olaraq, hər bir əlaqə üçün əgər əlaqə tərsinə çevrilməlidirsə sətirdə '1', əks halda '0' qoymalısınız. Birdən çox həll yolu varsa, onlardan hər hansı birini qaytara bilərsiniz.

Siz sistemə *find_reverse* funksiyasının icrasını ehtiva edən **monopoly.cpp** faylını göndərməlisiniz. Faylda proqramınız üçün lazım olan digər funksiyalar və kodlar ola bilər, lakin o, əsas funksiya olan *main*-dən **ibarət olmamalıdır**. Həmçinin, siz standart girişdən oxumağa və standart çıxışa yazmağa çalışmamalısınız!

Məhdudiyyətlər

- ♣ $3 \leq N \leq 5 \times 10^5$
- ♣ $3 \leq M \leq 1.5 \times 10^6$

Alt tapşırıqlar

Alt tapşırıq	Xal	N	M	Əlavə məhdudiyyətlər
1	0	–	–	Nümunə
2	15	≤ 7	≤ 21	–
3	40	$\leq 10^3$	$\leq 5 \times 10^3$	–
4	25	$\leq 10^5$	$\leq 5 \times 10^5$	–
5	20	$\leq 5 \times 10^5$	$\leq 1.5 \times 10^6$	–

Alt tapşırıq üçün xallar yalnız həmin alt tapşırığın bütün testləri uğurla keçdikdə verilir.

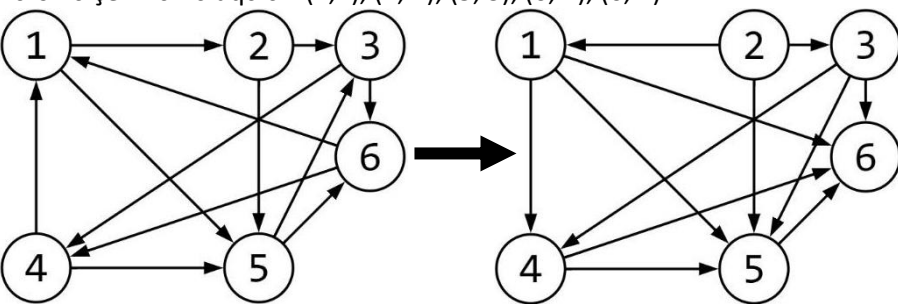
Lokal sınaq (Kompüterinizdə həllinizi test etmə)

Lokal sınaq üçün sizə **Lgrader.cpp** faylı verilir. Siz onu **monopoly.cpp** proqramınızla eyni qovluğa yerləşdirməli və **Lgrader.cpp** və **monopoly.cpp** fayllarını birlikdə kompayl etməlisiniz. Beləliklə, funksiyanızın düzgünlüyünü yoxlamaq üçün bir proqram hazırlamış olacaqsınız. Proqram standart girişdən aşağıdakı məlumat ardıcılığını tələb edəcək:

- birinci sətirdə: iki tam ədəd – xanaların sayı N və yeni lövhədəki əlaqələrin sayı M .
- növbəti M sətirdə: hər sətirdə iki tam ədəd x və y – hər bir istiqamətlənmiş əlaqə üçün başlanğıc və son xanalar.

Proqramın çıxışı tapdığınız ikili sətir olacaq.

Lokal sınaq nümunəsi

Giriş	Çıxış	İzah
6 12 1 2 1 5 2 3 2 5 3 4 3 6 4 1 4 5 5 3 5 6 6 1 6 4	100000101011	<p>Tərsinə çevrilən əlaqələr: (1,2), (4, 1), (5, 3), (6, 1), (6, 4).</p>  <p>Yuxarıdakı şəkil, birincisi, Deninin siyahısından istiqamətlənmiş əlaqələri olan ikincisi isə, çıxışda müvafiq '1'-lər ilə əlaqələrin istiqamətini tərsinə çevirdikdən sonra xanaları göstərir. Aydındır ki, başlanğıcda (1, 2), (2, 3), (3, 4) və (4, 1) istiqamətlənmiş əlaqələrdən istifadə edərək 1-ci xanadan başlaya bilərik və bu istiqamətlənmiş əlaqələr boyunca hərəkət edərək, yenidən ona qayıda bilərik. Yaranan lövhədə Deninin şərtinin ödəndiyini görmək olar.</p> <p>Başqa düzgün həll yolları da var: (4, 1), (5, 3), (6, 1), (6, 4) (1, 5), (2, 3), (2, 5), (3, 4), (3, 6), (4, 5), (5, 6)</p>