

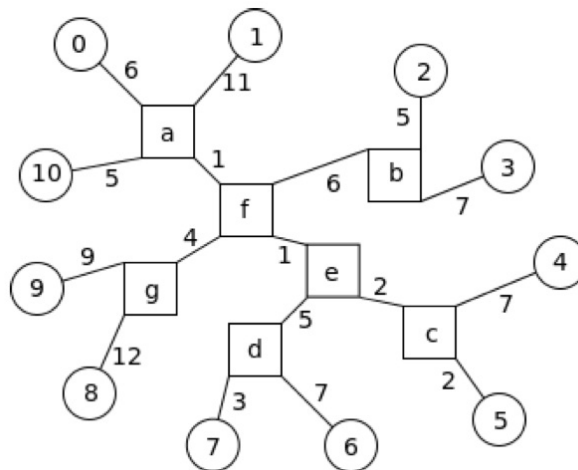
დაბები

მოცემულია N რაოდენობის დაბა (პატარა ქალაქი) ყაზახეთში, რომლებიც გადანომრილია 0 -დან $(N - 1)$ -მდე. გარდა ამისა, გვაქვს დაუდგენელი რაოდენობის დიდი ქალაქი, რომლებსაც ქვემოთ უბრალოდ “ქალაქი” ვუწოდოთ. დაბებს და ქალაქებს ერთობლივად ვუწოდოთ “დასახლებული პუნქტები”.

ყაზახეთის ყველა დასახლებული პუნქტი დაკავშირებულია ერთადერთი საგზაო ქსელით, რომელიც შედგენილია ორმხრივად მიმართული გზებისაგან. ყოველი გზა აერთებს ორ განსხვავებულ დასახლებულ პუნქტს და დასახლებული პუნქტების ყოველი წყვილი შეერთებულია არაუმეტეს ერთი პირდაპირი გზით. a და b დასახლებული პუნქტების ყოველი წყვილისათვის არსებობს ერთადერთი მარტივი მარშრუტი, რომლითაც შეიძლება მივიდეთ a -დან b -მდე (ანუ მარშრუტში ყოველი გზა გამოიყენება არაუმეტეს ერთხელ).

ცნობილია, რომ ყოველი დაბა პირდაპირ დაკავშირებულია ერთადერთ დასახლებულ პუნქტთან, ხოლო ყოველი ქალაქი პირდაპირ დაკავშირებულია 3 ან მეტ დასახლებულ პუნქტთან.

ქვემოთ, ნახაზზე მოცემულია ქსელი, რომელიც შედგება 11 დაბისა და 7 ქალაქისაგან. დაბები გამოსახულია, როგორც წრეები და აღნიშნულია მთელი რიცხვებით, ხოლო ქალაქები ნაჩვენებია, როგორც წრეები და აღნიშნულია ლათინური სიმბოლოებით.



ყოველი გზის სიგრძე გამოსახულია მთელი დადებითი რიცხვით. მანძილი ორ დასახლებულ პუნქტს შორის წარმოადგენს მათ შორის არსებული მარტივი მარშრუტის შემადგენელი გზების სიგრძეთა ჯამს.

ყოველი C ქალაქისათვის ჩვენ შეგვიძლია განვსაზღვროთ $r(C)$ სიდიდე, როგორც მანძილი დაბამდე, რომელიც ყველაზე მეტადაა დაშორებული ამ

ქალაქიდან. C ქალაქი წარმოადგენს *ჰაბს*, თუ მისი $r(C)$ უმცირესია ყველა ქალაქის იგივე მახასიათებელს შორის. მანძილი \mathbb{Z} -სა და \mathbb{Z} -შიდან ყველაზე დაშორებულ დაბას შორის აღვნიშნოთ R -ით. მაშასადამე, R არის უმცირესი $r(C)$ -ს მნიშვნელობებს შორის.

ზემოთ მოყვანილ მაგალითში უმცირესი დაბა a ქალაქიდან არის დაბა 8 და მანძილი მათ შორის ტოლია $r(a) = 1 + 4 + 12 = 17$. g ქალაქისთვისაც ასევე $r(g) = 17$ (ყველაზე დაშორებული დაბა g ქალაქისათვის არის დაბა 6). ერთადერთი \mathbb{Z} -ში მოცემულ მაგალითში არის ქალაქი f , რადგან $r(f) = 16$. შესაბამისად, ზემოთ ნახვენებ მაგალითში R არის 16.

\mathbb{Z} -ში წაშლა იწვევს ქსელის დაშლას რამდენიმე ბმულ ნაწილად. \mathbb{Z} -ში ითვლება “ბალანსირებულად”, თუ ეს ნაწილები შეიცავენ არაუმეტეს $\lfloor N/2 \rfloor$ დაბას (მიაქცევთ ყურადღება, რომ ქალაქები არ ითვლება). შევნიშნოთ, რომ $\lfloor x \rfloor$ აღნიშნავს უდიდეს მთელ რიცხვს, რომელიც არ აღემატება x -ს.

ჩვენს მაგალითში \mathbb{Z} -ს წარმოადგენს f ქალაქი. თუკი მას წავშლით, ქსელი დაიშლება 4 ბმულ ნაწილად, რომლებიც შეიცავენ შემდეგ დაბებს: $\{0, 1, 10\}$, $\{2, 3\}$, $\{4, 5, 6, 7\}$, და $\{8, 9\}$. ამ ნაწილებიდან არცერთი არ შეიცავს $\lfloor 11/2 \rfloor = 5$ დაბაზე მეტს, ამიტომ f ბალანსირებული \mathbb{Z} -ია.

ამოცანა

თავდაპირველად, ერთადერთი ინფორმაცია, რომელიც თქვენ გაქვთ დასახლებულ პუნქტთა ქსელის შესახებ, არის დაბების რაოდენობა N . თქვენ არ იცით ქალაქების რაოდენობა და ასევე არ იცით გზების განლაგება ქვეყანაში. თქვენ შეგიძლიათ ინფორმაციის მიღება შეკითხვების შედეგად, რომლებიც შეეხებიან მანძილს დაბათა წყვილს შორის.

თქვენი ამოცანაა, განსაზღვროთ:

- ყველა ქვეამოცანაში: R მანძილი.
- ქვეამოცანებში 3–დან 6–მდე: არის თუ არა ქსელში ბალანსირებული \mathbb{Z} -ია.

თქვენ უნდა მოახდინოთ ფუნქცია `hubDistance`-ს იმპლემენტაცია. გრადერი შეაფასებს მრავალჯერად ტესტებს ერთი გაშვებისას. ტესტების რაოდენობა ერთი გაშვებისას არ აღემატება 40-ს. თითოეული ტესტისათვის გრადერი გამოიძახებს თქვენს `hubDistance` ფუნქციას მხოლოდ ერთხელ. თქვენმა ფუნქციამ უნდა მოახდინოს ყველა საჭირო მონაცემის ინიციალიზაცია ყოველი გამოძახებისას.

- `hubDistance(N, sub)`
 - N : დაბათა რაოდენობა.
 - `sub`: ქვეამოცანის ნომერი (ახსნილია ქვეამოცანების განყოფილებაში).
 - თუ `sub` არის 1 ან 2, ფუნქციამ უნდა დააბრუნოს R ან $-R$
 - თუ `sub` არის 2–ზე მეტი, მაშინ თუკი არსებობს ბალანსირებული \mathbb{Z} -ია, ფუნქციამ უნდა დააბრუნოს R , წინააღმდეგ შემთხვევაში უნდა დააბრუნოს $-R$.

თქვენმა `hubDistance` ფუნქციამ უნდა მიიღოს ინფორმაცია ქსელის შესახებ გრაფერის ფუნქციისაგან `getDistance(i, j)`. ეს ფუნქცია აბრუნებს მანძილს i და j დაბებს შორის. შევნიშნოთ, რომ თუ i და j ტოლია, ფუნქცია აბრუნებს 0-ს. ის ასევე 0-ს აბრუნებს, როცა ფუნქციის არგუმენტები არასწორია.

ქვეამოცანები

ყოველი ტესტისათვის:

- N -ის მნიშვნელობა მოთავსებულია 6-დან 110-მდე მათი ჩათვლით.
- მანძილი ორ დაბას შორის მოთავსებულია 1-დან 1,000,000-მდე მათი ჩათვლით.

თქვენი პროგრამის მიერ გაკეთებული შეკითხვების რაოდენობა შეზღუდულია და მათი რაოდენობა აღწერილია ქვემოთ მოყვანილ ცხრილში. თუკი თქვენი პროგრამა გადააჭარბებს დაწესებულ ლიმიტს, მისი მუშაობა შეწყდება და პასუხი არასწორად ჩაითვლება.

ქვეამ.	ქულა	შეკითხვების რაოდენობა	ბალანსირებული ჰაბის ძებნა	დამატებითი შეზღუდვა
1	13	$\frac{N(N-1)}{2}$	NO	არა
2	12	$\lceil \frac{7N}{2} \rceil$	NO	არა
3	13	$\frac{N(N-1)}{2}$	YES	არა
4	10	$\lceil \frac{7N}{2} \rceil$	YES	ყოველი ქალაქი დაკავშირებულია ზუსტად 3 დასახლებულ პუნქტთან
5	13	$5N$	YES	არა
6	39	$\lceil \frac{7N}{2} \rceil$	YES	არა

მიაქციეთ ყურადღება, რომ $\lceil x \rceil$ აღნიშნავს უმცირეს მთელ რიცხვს, რომელიც მეტია ან ტოლი x -ზე.

სანიმუშო გრაფერი

მიაქციეთ ყურადღება, რომ ქვეამოცანის ნომერი წარმოადგენს შესატანი მონაცემის ნაწილს. სანიმუშო გრაფერი ცვლის საკუთარ მოქმედებას ქვეამოცანის ნომრის მიხედვით.

სანიმუშო გრაფერი კითხულობს მონაცემებს `towns.in` ფაილიდან შემდეგი ფორმატით:

- სტრიქონი 1: ქვეამოცანის ნომერი და ტესტების რაოდენობა.
- სტრიქონი 2: N_1 – დაბების რაოდენობა პირველი ტესტისათვის.
- მომდევნო N_1 რაოდენობის სტრიქონში: j -ური რიცხვი ($1 \leq j \leq N_1$) i -ურ სტრიქონში ($1 \leq i \leq N_1$) აღნიშნავს მანძილს $i - 1$ და $j - 1$ დაბებს შორის.
- მომდევნო ტესტები ისევეა აღწერილი, როგორც პირველი ტესტი.

ყოველი ტესტისათვის სანიმუშო გრაფერი ბეჭდავს `hubDistance`-ის

დაბრუნებულ მნიშვნელობას და გამოძახებათა რაოდენობას.

შესატანი ფაილი ზემოთ აღწერილი მაგალითისათვის:

```
1 1
11
0 17 18 20 17 12 20 16 23 20 11
17 0 23 25 22 17 25 21 28 25 16
18 23 0 12 21 16 24 20 27 24 17
20 25 12 0 23 18 26 22 29 26 19
17 22 21 23 0 9 21 17 26 23 16
12 17 16 18 9 0 16 12 21 18 11
20 25 24 26 21 16 0 10 29 26 19
16 21 20 22 17 12 10 0 25 22 15
23 28 27 29 26 21 29 25 0 21 22
20 25 24 26 23 18 26 22 21 0 19
11 16 17 19 16 11 19 15 22 19 0
```

ეს ფორმატი ძალზე განსხვავდება გზათა სიისაგან. თქვენ უფლება გაქვთ შეცვალოთ სანიმუშო გრაფერი, ისე რომ გამოიყენოთ სხვა შესატანი ფორმატი.