FAKULTAS ILMU KOMPUTER

CSGE602040 - Struktur Data dan Algoritma
Semester Gasal - 2021/2022
TP 3 (rev 1)
Deadline: Sabtu, 4 Desember 2021, 23.55 WIB

`Dinamika Shinomiya Enterprises

Deskripsi

Shinomiya enterprises adalah perusahaan terbesar di dunia yang dipimpin oleh Kaguya Shinomiya sebagai CEO. Perusahaan memiliki banyak karyawan dengan program-program yang pro-karyawan sehingga karyawan betah bekerja disini. Pada mulanya ada N karyawan **yang dikodekan dengan angka 1 hingga N**. Karyawan i memiliki pangkat P_i, dimana **nilai lebih besar menyatakan pangkat lebih tinggi**. Karyawan senang menjalin pertemanan. Pada mulanya, ada M pertemanan antar karyawan. Pertemanan ke-i dinyatakan dengan U_i dan V_i yang berarti karyawan U_i dan V_i berteman **secara dua arah**. Kantor sangat dinamis dan ada 2 kejadian yang dapat mengubah kondisi kantor, yaitu:

- **TAMBAH U_i V_i**: Karyawan U_i dan karyawan V_i mulai berteman
- **RESIGN U_i**: karyawan U_i keluar dari Shinomiya enterprises :(

> Didefinisikan **network** dari U sebagai daftar semua karyawan yang dapat di *reach-out* oleh U, baik karena berteman secara langsung, atau melalui teman-temannya. Perhatikan bahwa U sendiri tidak termasuk dalam **network** dari U. Suatu karyawan V dikatakan dapat di *reach-out* oleh U jika dan hanya jika mereka berteman, atau ada beberapa karyawan perantara $K_1\ K_2\ K_3\ K_4 \ldots\ K_x$ dimana U berteman dengan K_1, K_i berteman dengan K_{i+1} (untuk semua $1 \leq i < x$), dan K_x berteman dengan V.

Penasaran dengan dinamika kantornya, Kaguya terkadang meminta divisi Data Science untuk menganalisis beberapa ukuran kuantitatif penting yang dirasa perlu sebagai pertimbangan dalam membuat program-program internal. Saat ini, Kaguya dapat meminta salah satu dari lima nilai kuantitatif berikut:

CARRY U: Kaguya ingin tahu siapa yang biasa membantu karyawan U. Dia ingin mengetahui pangkat tertinggi diantara **semua teman karyawan U**. Secara khusus, **nilai ini 0 apabila U tidak memiliki teman**.

BOSS U: Kaguya ingin mengetahui pangkat tertinggi diantara semua karyawan pada **network** U. Secara khusus, **nilai ini 0 apabila U tidak memiliki teman**.

SEBAR U V: Penyampaian Informasi dengan cepat sangat krusial di perusahaan raksasa seperti Shinomiya enterprises. Kaguya ingin mengetahui, jika karyawan U ingin mengirimkan pesan ke karyawan V, berapakah **banyaknya karyawan perantara minimal** yang diperlukan? Seorang karyawan dapat mengirimkan atau meneruskan pesan ke karyawan lain dengan dua cara berikut:

- Meneruskan pesan ke temannya.

- Meneruskan pesan ke karyawan lain **dengan pangkat sama** via telepati. Karyawan penerima tidak harus berteman dengan pengirim/penerus pesan.

Secara khusus, **nilai ini -1** apabila U tidak mungkin mengirimkan pesannya ke V.

SIMULASI: Yang namanya manusia pasti bisa minder. Kaguya merasa **karyawan yang semua temannya memiliki pangkat sama atau lebih tinggi dari dia** sangat rentan minder sehingga dikategorikan sebagai **karyawan rentan**. Kaguya ingin tim Data Science mencari tahu seberapa rentan perusahaannya. Tim Data Science akan melakukan simulasi berikut:

1. Buatlah daftar **karyawan rentan**.
2. Anggap karyawan-karyawan tersebut resign.
3. Langkah (2) mungkin menyebabkan beberapa karyawan lain menjadi **karyawan rentan**. Ulangi langkah (1) dan (2) sampai tidak ada karyawan rentan.

Tim Data Science akan melaporkan **banyaknya karyawan yang tersisa di perusahaan** setelah simulasi berakhir. Perhatikan bahwa ini **hanya simulasi**, sehingga tidak mempengaruhi hubungan pertemanan, maupun status pekerjaan karyawan di dunia sebenarnya.

NETWORKING: Dari SEBAR, Kaguya sadar bahwa membangun jaringan yang kuat sangat vital dalam perusahaan. Kini, Kaguya ingin agar untuk setiap karyawan perusahaan, **network-**nya berisi **semua karyawan lain di perusahaan**. Untuk memfasilitasi ini, Kaguya akan mengadakan program networking sebagai berikut:

1. Pilih dua buah karyawan U dan V yang **tidak berteman**
2. Buatlah mereka berteman dengan berbagai cara. **Usaha** yang diperlukan untuk membuat karyawan **berpangkat** X dan karyawan **berpangkat** Y menjadi teman adalah $|X - Y|$.
3. Ulangi langkah (1) dan (2) selama kondisi akhir yang diinginkan belum terpenuhi.

Kaguya ingin mengetahui **total usaha minimal** yang diperlukan untuk mencapai kondisi akhir yang ia inginkan.

Anda sebagai anggota tim Data Science akan mengerjakan semua permintaan Kaguya-sama.

Format Masukan

Baris pertama berisi 3 buah bilangan N M Q, banyak karyawan mula-mula, banyak pertemanan mula-mula, dan banyaknya kejadian atau pertanyaan Kaguya.

Baris berikutnya berisi N buah bilangan, dimana bilangan ke-i menyatakan P_i, pangkat dari karyawan ke-i

M baris berikutnya masing-masing berisi dua buah bilangan $U_i V_i$, yang menyatakan bahwa karyawan U_i dan V_i berteman.

Q baris berikutnya masing-masing berisi sebuah perubahan atau pertanyaan. Entri pertama berisi **kode** dari perubahan/pertanyaan tersebut, diikuti dengan masukan valid untuk perubahan/pertanyaan tersebut.
Berikut kode dari perubahan/pertanyaan pada soal:

Perubahan/Pertanyaan	Kode
TAMBAH	1
RESIGN	2
CARRY	3
BOSS	4
SEBAR	5
SIMULASI	6
NETWORKING	7

Format Keluaran

Untuk setiap pertanyaan Kaguya, keluarkan sebuah baris berisi jawaban yang diminta.

Batasan

Soal ini memiliki 4 kelompok test-case. Silahkan lihat **informasi tambahan test-case** untuk keterangan lebih lanjut. Untuk setiap kelompok, berlaku:

- $1 \leq N \leq 100.000$
- $0 \leq M \leq 200.000$
- $1 \leq P_i \leq N$
- Untuk semua pertemanan awal dan TAMBAH: $1 \leq U_i, V_i \leq N, U_i \neq V_i$. Dijamin U_i dan V_i belum berteman dan tidak resign saat pertemanan terbentuk
- Untuk semua RESIGN, dijamin U_i belum resign sebelumnya.
- DIjamin tidak semua karyawan RESIGN.

Batasan Kelompok 1

- Perubahan/pertanyaan yang mungkin hanya: TAMBAH, RESIGN, CARRY, SIMULASI
- $1 \leq Q \leq 100.000$
- Semua karyawan memiliki pangkat yang berbeda
- Dijamin karyawan yang ditanyakan saat CARRY belum resign

Batasan Kelompok 2

- Perubahan/pertanyaan yang mungkin hanya: NETWORKING
- $Q = 1$

Batasan Kelompok 3

- Perubahan/pertanyaan yang mungkin hanya: TAMBAH, RESIGN, SEBAR
- $1 \leq Q \leq 200$
- Dijamin dua karyawan yang terlibat dalam SEBAR belum resign.

Batasan Kelompok 4

- Perubahan/pertanyaan yang mungkin hanya: BOSS
- $1 \leq Q \leq 100.000$

Contoh Masukan 1

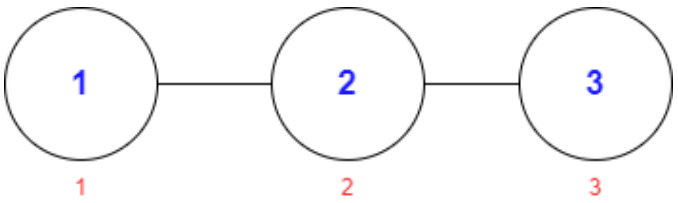
```
3 2 5
1 2 3
1 2
2 3
3 1
2 2
3 1
1 1 3
3 1
```

Contoh Keluaran 1

```
2
0
3
```

Penjelasan 1

Ilustrasi awal setelah membangun graf pertemanan:
Biru menandakan nomor karyawan, merah menandakan pangkat



1. **3 1 (CARRY 1)**
   Pada CARRY pertama, teman dari karyawan 1 yang berpangkat paling tinggi adalah karyawan 2 dengan pangkat 2.
2. **2 2 (RESIGN 2)**
   Menghapus karyawan 2 dari pertemanan.
3. **3 1 (CARRY 1)**
   Setelah karyawan 2 RESIGN, pada CARRY kedua karyawan 1 tidak punya teman sehingga output 0
4. **1 1 3 (TAMBAH 1 3)**
   Membuat pertemanan antara karyawan 1 dan karyawan 3
5. **3 1 (CARRY 1)**
   Setelah karyawan 1 dan 3 berteman, CARRY ketiga dijalankan dan menghasilkan 3 karena ranking teman karyawan ke-1 yang paling tinggi adalah karyawan 3 dengan pangkat 3

Contoh Masukan 2

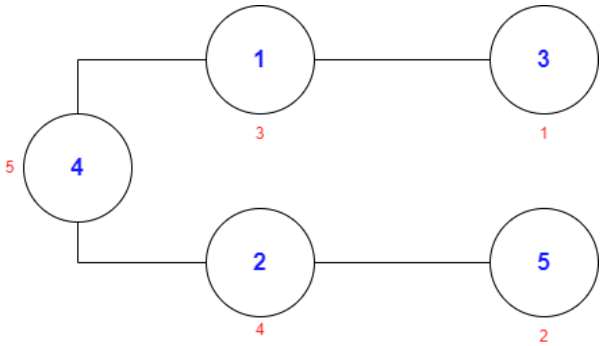
```
5 4 5
3 4 1 5 2
4 1
3 1
5 2
2 4
6
1 3 5
3 3
2 4
6
```

Contoh Keluaran 2

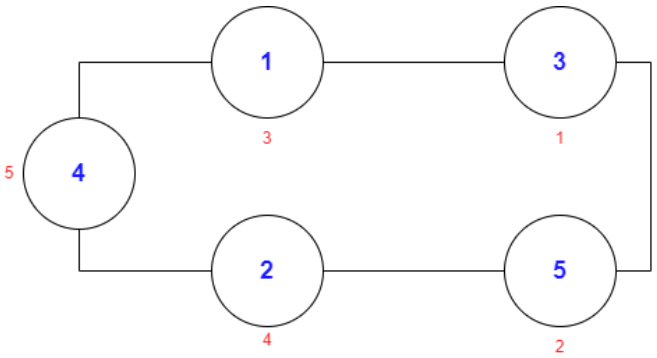
```
1
3
2
```

Penjelasan 2

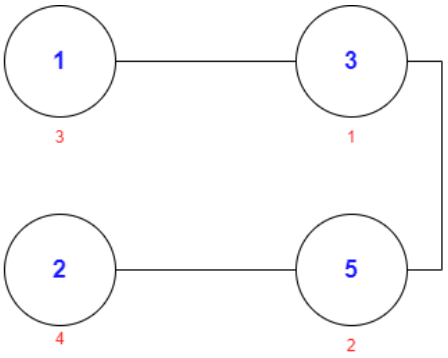
Ilustrasi awal setelah membangun graf pertemanan:



1. **6 (SIMULASI)**
   Pada simulasi pertama:
   - Pada mulanya karyawan 3 dan 5 rentan, sehingga mereka dianggap resign
   - Akibatnya, karyawan 2 dan 1 menjadi rentan, sehingga mereka dianggap resign
   - Tidak lagi karyawan yang rentan. Sehingga pada akhirnya, hanya **1 karyawan** yang tersisa, yaitu karyawan 4.
2. **1 3 5 (TAMBAH 3 5)**
   Membuat pertemanan antara karyawan 3 dan 5. Ilustrasi graph setelah penambahan pertemanan:



3. **3 3 (CARRY 3)**
   Teman karyawan 3 yang memiliki pangkat tertinggi adalah karyawan 1
4. **2 4 (RESIGN 4)**
   Karyawan 4 telah keluar dari Shinomiya enterprises. Ilustrasi graph setelah resign:



5. **6 (SIMULASI)**
   Pada simulasi pertama:
   - Pada mulanya karyawan 3 rentan sehingga dianggap resign. Hal tersebut memecah network menjadi dua.
   - Akibatnya, karyawan 5 dianggap resign.
   - Tidak lagi karyawan yang rentan. Sehingga pada akhirnya, ada **2 karyawan** yang tersisa, yaitu karyawan 1 dan 2.

Contoh Masukan 3

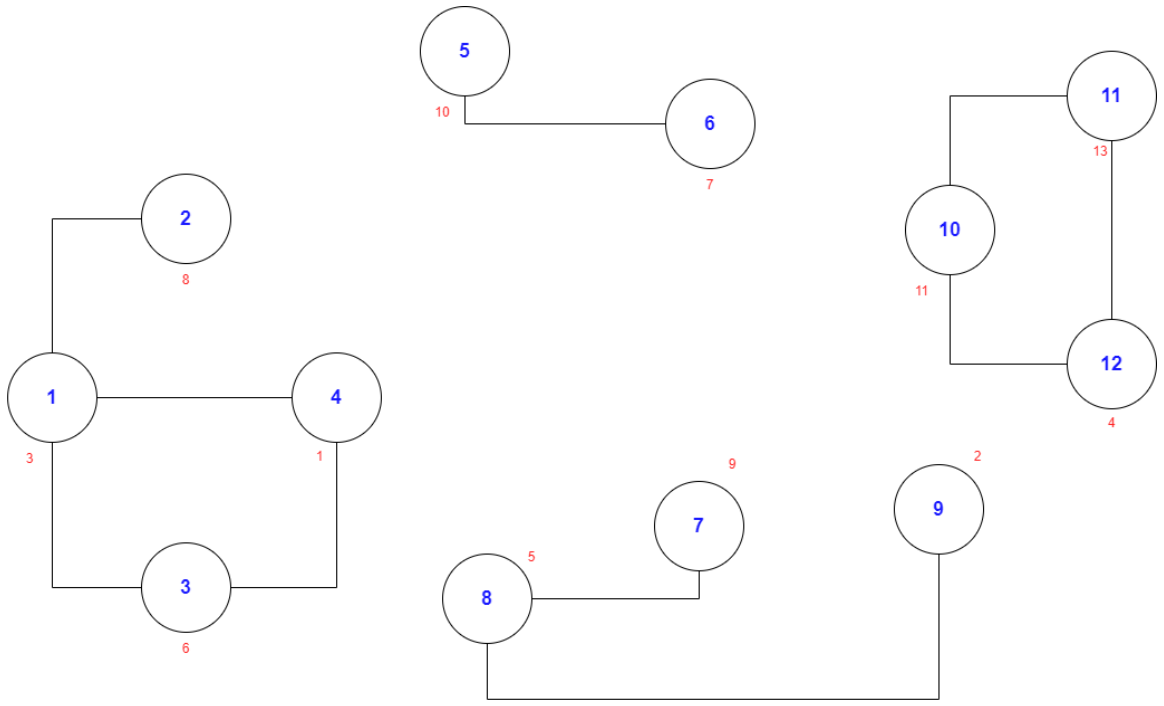
```
12 10 1
3 8 6 1 10 7 9 5 2 11 13 4
1 2
1 4
1 3
3 4
5 6
7 8
8 9
10 11
11 12
10 12
7
```

Contoh Keluaran 3

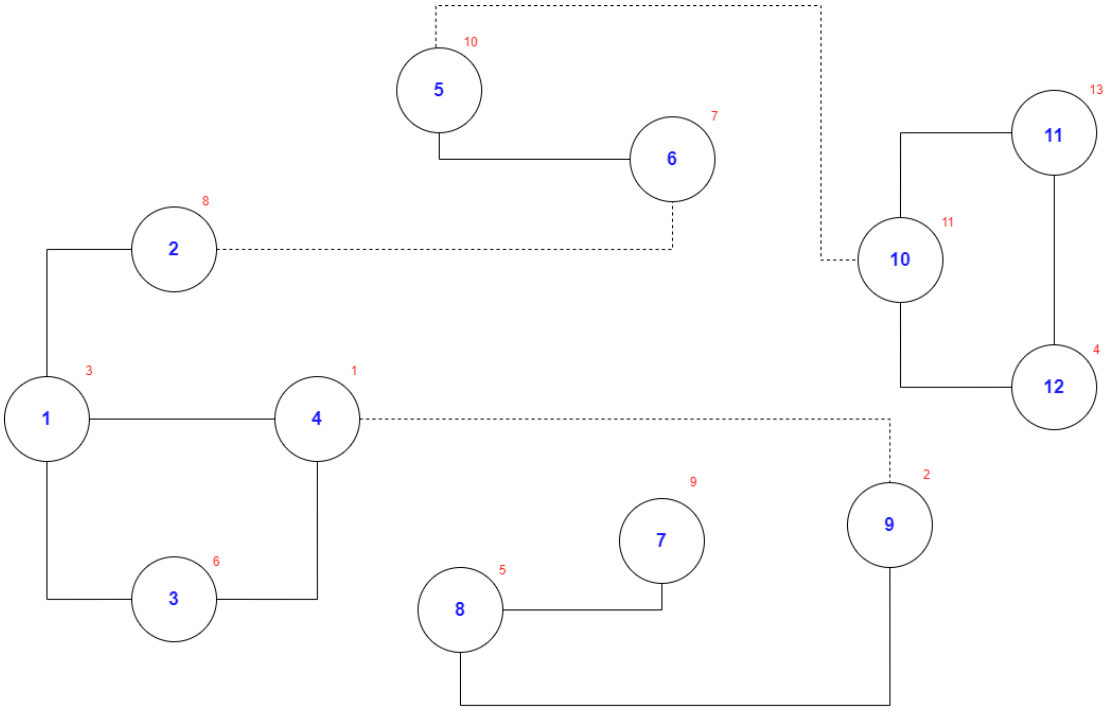
3

Penjelasan 3

Ilustrasi awal setelah membangun graf pertemanan :



Setelah NETWORKING dijalankan, maka hasil graphnya akan menjadi seperti ini (garis putus-putus merupakan hasil NETWORKING) :



Bisa dilihat bahwa usaha totalnya adalah :

- Karyawan 2 berteman dengan karyawan 6 → |8 - 7| = 1 usaha
- Karyawan 5 berteman dengan 10 → |10 - 11| = 1 usaha
- Karyawan 4 berteman dengan karyawan 9 → |1 - 2| = 1 usaha

Maka **total usaha minimal** yang diperlukan adalah 3
Perhatikan bahwa ada banyak pilihan lain untuk melakukan NETWORKING usaha minimal, seperti membuat karyawan 1 dan karyawan ke-12 berteman dan membiarkan karyawan 5 dan 10 tidak berteman.

Contoh Masukan 4

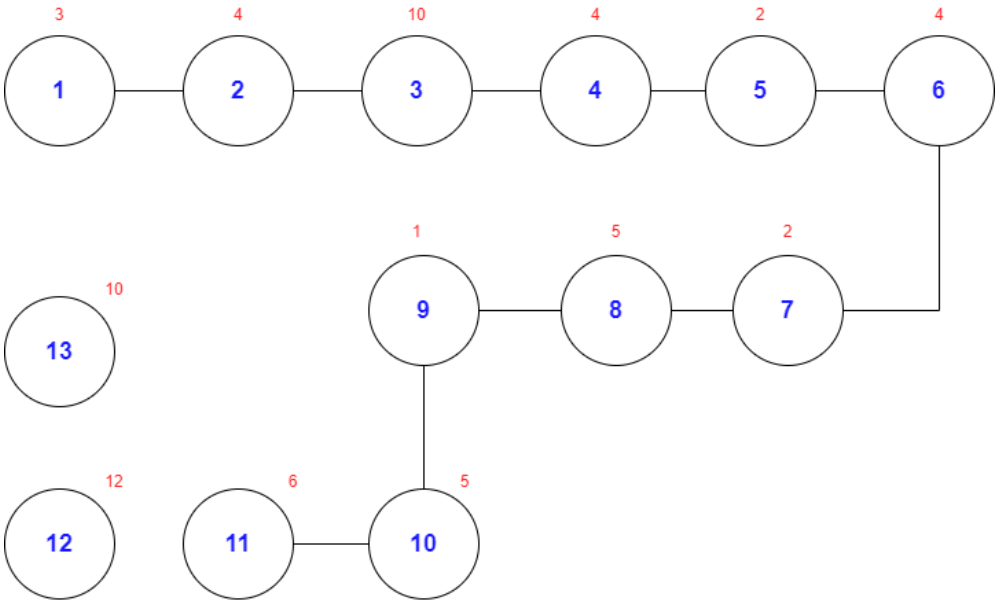
```
13 10 3
3 4 10 4 2 4 2 5 1 5 6 12 10
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
10 11
5 1 11
5 1 12
5 1 13
```

Contoh Keluaran 4

```
5
-1
2
```

Penjelasan 4

Ilustrasi awal setelah membangun graf pertemanan:

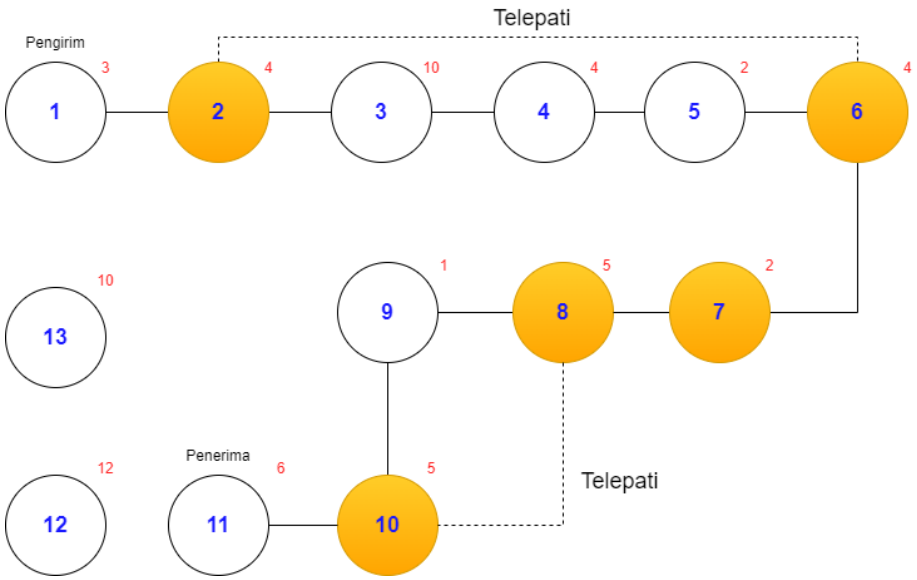


1. **5 1 11 (SEBAR 1 11)**

   Berikut Alur pengirimannya saat SEBAR:

   1. Karyawan 1 meneruskan pesan ke karyawan 2 (teman)
   2. Karyawan 2 mengirim pesan via telepati ke karyawan 6. Perhatikan bahwa karyawan 2 bisa saja telepati ke karyawan 4, atau meneruskan pesan ke karyawan 3 (temannya). Namun, paling optimal adalah telepati ke karyawan 6.
   3. Karyawan 6 meneruskan pesan ke karyawan 7 (teman)
   4. Karyawan 7 meneruskan pesan ke karyawan 8 (teman). Perhatikan bahwa bisa saja karyawan 7 telepati ke karyawan 5 karena pangkat sama. Namun, paling optimal adalah meneruskan pesan ke karyawan 8.
   5. Karyawan 8 telepati ke karyawan 10.
   6. Karyawan 10 meneruskan pesan ke karyawan 11.

   Jadi, minimal ada **5** karyawan perantara untuk mengirimkan pesan dari karyawan 1 ke karyawan 11. Dibawah merupakan karyawan-karyawan perantara dimana node kuning menyatakan karyawan yang digunakan sebagai perantara.



2. **5 1 12 (SEBAR 1 12)**
   Kalau kita lihat ilustrasi diatas, pesan dari karyawan 1 ke karyawan 12 tidak dapat dicapai melalui persebaran teman. Karyawan 12 juga memiliki pangkat berbeda dengan karyawan lain, sehingga tidak ada cara untuk mengirimkan pesan via telepati. Jadi, output dari perintah ini adalah **-1**.
3. **5 1 13 (SEBAR 1 13)**
   Berikut Alur pengirimannya saat SEBAR:
   1. Karyawan 1 meneruskan pesan ke karyawan 2 (teman)
   2. Karyawan 2 meneruskan pesan ke karyawan 3 (teman)
   3. Karyawan 3 mengirim pesan via telepati ke karyawan 13. Perhatikan bahwa karyawan 3 memiliki pangkat yang sama dengan karyawan 13, sehingga telepati dapat dilakukan.

   Jadi, minimal ada **2** karyawan perantara untuk mengirimkan pesan dari karyawan 1 ke karyawan 13.

Contoh Masukan 5

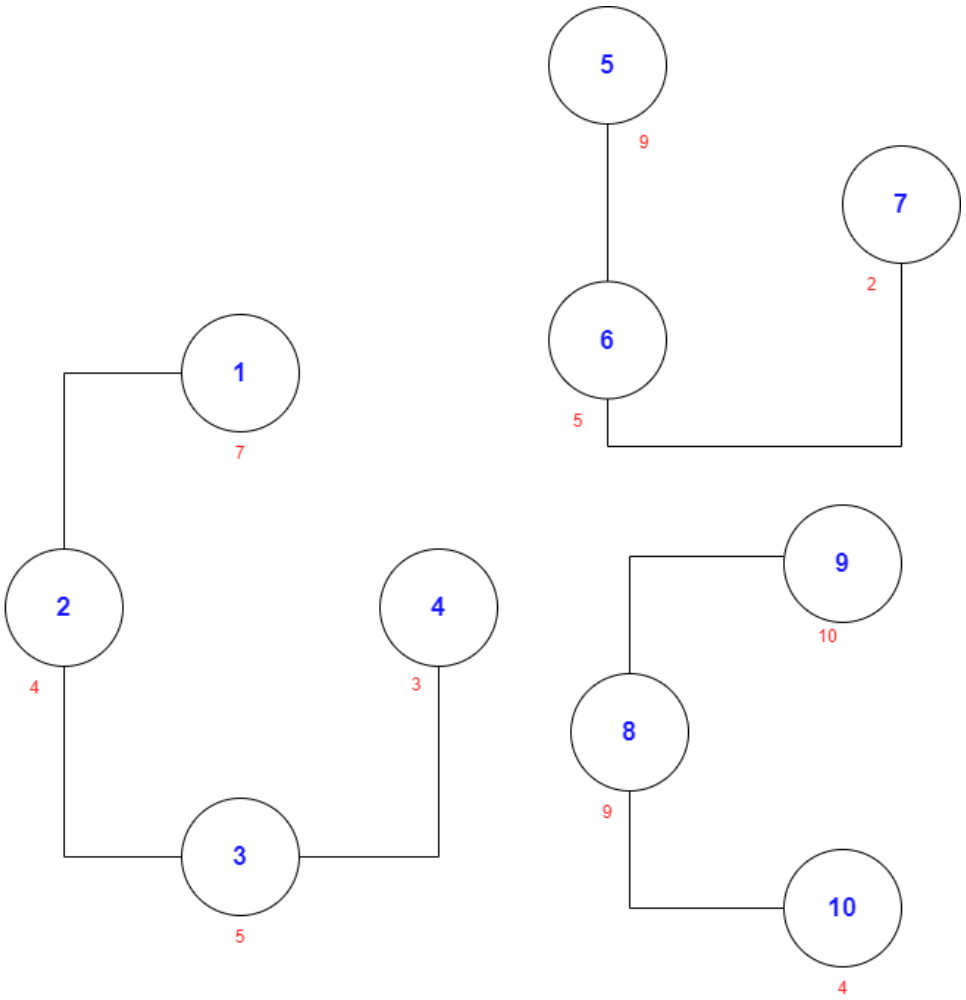
```
10 7 3
7 4 5 3 9 5 2 9 10 4
1 2
2 3
3 4
5 6
6 7
8 9
8 10
4 2
4 7
4 8
```

Contoh Keluaran 5

```
7
9
10
```

Penjelasan 5

Ilustrasi awal setelah membangun graf pertemanan :



Pada pertanyaan BOSS pertama, network dari karyawan 2 yang mempunyai pangkat tertinggi adalah karyawan 1 yang mempunyai pangkat 7.
Pada pertanyaan BOSS kedua, network dari karyawan 7 yang mempunyai pangkat tertinggi adalah karyawan 5 yang mempunyai pangkat 9.
Pada pertanyaan BOSS ketiga, network dari karyawan 8 yang mempunyai pangkat tertinggi adalah karyawan 9 yang mempunyai pangkat 10.

Keterangan Tambahan

Struktur data bawaan java yang diperbolehkan pada TP ini hanya **primitive array, ArrayList, Stack, Queue, Vector, LinkedList, dan ArrayDeque**. Penggunaan struktur data bawaan lain, seperti **HashMap, HashSet, TreeMap** dilarang dan akan mendapatkan **penalti 20%**.

TIdak diperbolehkan untuk mengimplementasikan BST sendiri, seperti AVL, treap, RB-tree. Tidak diperbolehkan untuk menggunakan algoritma atau struktur data **tingkat lanjut** yang belum atau tidak diajarkan pada SDA, seperti hash table, Dinic algorithm, Hungarian algorithm, dan ear decomposition. Pelanggaran akan mendapatkan **penalti 20%**

Tidak diperbolehkan untuk menggunakan fungsionalitas sorting bawaan java, seperti **collections.Sort**. Anda wajib mengimplementasikan sorting sendiri jika solusinya memerlukan sorting. Pelanggaran akan mendapatkan **penalti 15%**.

Anda bebas untuk memakai seluruh algoritma graf yang sudah atau akan dipelajari di SDA, seperti Dijkstra, Kruskal (dan struktur data union-find), Prim dan DFS/BFS.

Informasi Tambahan Test-case

Kelompok	**Jenis Pertanyaan/Perubahan**	**Nomor TC**
Kelompok 1	TAMBAH, CARRY	1 - 5
	TAMBAH, CARRY, RESIGN	6 - 13
	TAMBAH, CARRY, RESIGN, SIMULASI	14 - 33
Kelompok 2	NETWORKING	34 - 43
Kelompok 3	SEBAR Semua pangkat karyawan berbeda	44 - 51
	TAMBAH, RESIGN, SEBAR Semua pangkat karyawan berbeda	52 - 54
	TAMBAH, RESIGN, SEBAR Hanya ada dua pangkat berbeda di kantor	55 - 61
	TAMBAH, RESIGN, SEBAR	62 - 75
Kelompok 4	BOSS	76 - 100

Changelog

Revisi 1:

- Mengubah batasan $1 \leq M \leq 200.000$ menjadi $0 \leq M \leq 200.000$
- Menambahkan batasan pada kelompok 1 bahwa semua karyawan memiliki pangkat yang berbeda
- Mengganti pangkat 11 dan 13 menjadi pangkat 9 dan 10 di contoh masukan 5 menjadi karena melanggar batasan yang diberikan.