

PSA2: DN4

*Ne, to ni piramidna shema,
samo še dva pripeljite.*

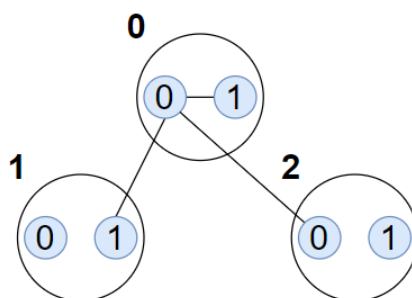
direktor piramidne sheme

V podjetju (raje ga ne imenujmo) vlada strogi red. Vsaka od n enot v podjetju – označimo jih z e , kjer je $0 \leq e < n$ – ima natanko eno sebi nadrejeno enoto. Izjema je enota 0, v kateri so zaposleni vrhovni šefi. Več enot ima lahko isto nadrejeno enoto. Ker nižja številka enote pomeni večji prestiž, velja: če je e' nadrejena enoti e , potem velja $e' < e$.

V vsaki enoti dela natanko k ljudi. Zaposlenega torej določa par $(e, i) \in [n] \times [k]$, kjer je $[a] = \{0, 1, \dots, a-1\}$. Zaposlena se morda poznata le v primerih, ko

- delata v isti enoti (ker sedita v isti pisarni), ali
- ko je eden od njiju neposredno nadrejen drugemu (ker podrejeni plačuje provizijo nadrejenemu).

Primer podjetja z $n = 3$ enotami in $k = 2$ zaposlenima v vsaki enoti je prikazan na spodnji sliki:



Enota 0 je neposredno nadrejena enotama 1 in 2. To na skici ni označeno, lahko pa o tem sklepamo, saj zaposleni $z_1 = (0, 0)$ pozna osebe $z_2 = (0, 1)$, $z_3 = (1, 1)$ in $z_4 = (2, 0)$. Poznanstva $\{z_1, z_2\}$, $\{z_1, z_3\}$ in $\{z_1, z_4\}$ so tudi edina tri poznanstva. Poznanstva so seveda **neusmerjena**.

Naloga A (10 točk)

Vodstvo ugotavlja, da je nivo poznavanja med zaposlenimi precej nizek, zato se je ob koncu leta odločilo organizirati preživetveni turnir v naravi, ki pa

se ga ne bodo mogli udeležiti vsi! Da ne bi prišlo do sklepanja zavezništev pred tekmo, se lahko turnirja udeleži največ ena oseba iz vsakega poznanstva $\{z, z'\}$. Tisti, ki ne poznajo nikogar, nimajo omejitev.

Katero je največje število ljudi, ki se lahko udeleži turnirja?

Koda in podatki

Za kodo velja tako kot do sedaj, le da je tokrat malo bolj radodarna, saj pokaže, kako definirati zgoščevalno funkcijo za pare `std::pair<int, int>`, če jo boste slučajno potrebovali (napake, ki jih sicer `c++` izpljune ob uporabi `std::unordered_map`, kjer so kjuči pari, niso ravno informativne).

Tokrat sta podnalogi le dve: testna A0 in dejanska A1. Testni primeri v vhodni datoteki (končnica `.in`) so podani z uvodno vrstico **GRAF n k p** , ki ji sledi p vrstic, ki opisujejo poznanstva v podjetju. Vsako poznanstvo je predstavljeno s četverico **e i e' i'** , ki pove, da se osebi (e, i) in (e', i') poznata.

Pozor! Včasih je povezav tako malo, da se hierarhične strukture podjetja ne da enolično določiti. Premislite, kaj storiti v tem primeru.

Število enot n gre do 1000, število zaposlednih na enoto k pa je omejeno z 8. Zgornja meja za k je sumljivo nizka :)

Skupen dovoljen čas za vsako od nalog A0 in A1 je 10 minut, zato ker uradna rešitev potrebuje 72 s in ne želimo izgubljati časa z dinamičnim prilagajanjem meje (naivna rešitev je tako ali tako veliiiiiiiiko prepočasna).

Primer

Vsebina `A0.in` in `A0.out` je prikazana spodaj:

```
A0.in:
GRAF 3 2 3
0 0 0 1
0 0 1 1
0 0 2 0
```

```
A0.out:
5
```

To je natanko graf iz primera na skici zgoraj. Vseh šest zaposlenih se turnirja ne more udeležiti, če pa $(0, 0)$ ostane doma, pa se ga vsi preostali lahko.