

Krokodillide maa-alune linn

Arheoloog Benjamas uuris krokodillide maa-alust linna ja püüab nüüd sellest välja saada. Linn koosneb **N** kambrist ja **M** koridorist. Iga koridor ühendab omavahel kaht erinevat kambrist ja erinevad koridorid ühendavad erinevaid kambripaare. Erinevate koridoride läbimiseks võib kuluda erinev hulk aega. **N** kambrist **K** on väljapääsud maale. Benjamas alustab kambrist number **0** ja tahab võimalikult kiiresti väljapääsuni jõuda.

Krokodillide valveülem tahab Benjamase põgenemist takistada. Ta juhib oma koopast salauksi ja võib nende abil sulgeda ükskõik millise koridori, aga ainult ühe koridori korraga. See tähendab, et kui ta mõne koridori sulgeb, siis eelmine suletud koridor avaneb.

Benjamase olukord on seega järgmine. Iga kord, kui ta on mingis kambris ja tahab sealt edasi liikuda, võib valveülem valida ühe koridori ja selle sulgeda. Benjamas saab seejärel valida ühe sulgemata koridori ja seda mööda mõnda teise kambrisse liikuda. Kui Benjamas on juba koridoris sees, seda enam sulgeda ei saa. Kui Benjamas jõuab järgmisse kambrisse, võib valveülem jälle sulgeda ühe sellest väljuva koridori (ka selle, mille kaudu Benjamas just saabus) ja nii edasi.

Benjamas tahab, et tal oleks selge põgenemisplaani. Täpsemalt tahab ta juhiseid, mida teha, kui ta jõuab ühte või teise kambrisse. Olgu **A** üks kambritest. Kui **A** on ühtlasi väljapääs, siis pole muidugi mingeid juhiseid vaja, sest Benjamas saab lihtsalt krokodillide linnast välja astuda. Vastasel korral peab juhis olema ühel kahest järgmisest kujust:

- ♣ "Kui sa jõuad kambrisse **A**, mine edasi koridori, mis viib kambrisse **B**. Kui see koridor on suletud, mine selle asemel koridori, mis viib kambrisse **C**."
- ♣ "Kamber **A** ei puutu asjasse, seda plaani järgides ei jõua sa kunagi sinna."

Pange tähele, et mõnel juhul (näiteks, kui Benjamas jääb plaani järgides tsüklisse) on valveülemal võimalik Benjamasel linnast lahkumist takistada. Nimetame põgenemisplaani *heaks*, kui selle järgimine tagab Benjamasele linnast väljumise lõpliku aja jooksul, vaatamata sellele, mida valveülem teeb. Olgu mingi hea plaani korral **T** minimaalne selline aeg, et Benjamas pääseb *igal juhul* linnast välja hiljemalt aja **T** jooksul. Sel juhul ütleme, et selle *hea plaani ajakulu* on **T**.

Ülesanne

Kirjutada funktsioon `travel_plan(N,M,R,L,K,P)`, mille parameetrid on järgmised:

- ♣ **N** – kambrite arv; kambrid on nummerdatud $0 \dots N-1$.
- ♣ **M** – koridoride arv; koridorid on nummerdatud $0 \dots M-1$.
- ♣ **R** – kahemõõtmeline täisarvude massiiv, mis kirjeldab koridore; iga $0 \leq i < M$ korral ühendab koridor **i** kambreid **R[i][0]** ja **R[i][1]**; on teada, et mingid kaks erinevat koridori ei ühenda sama kambripaari.
- ♣ **L** – ühemõõtmeline täisarvude massiiv, milles on koridoride läbimiseks kuluvad ajad; iga $0 \leq i < M$ korral on $1 \leq L[i] \leq 1\,000\,000\,000$ aeg, mis kulub koridori **i** läbimiseks.
- ♣ **K** – väljapääsude arv; võib eeldada, et $1 \leq K < N$.
- ♣ **P** – ühemõõtmeline täisarvude massiiv, mille elemendid (kõik unikaalsed) kirjeldavad väljapääse; iga $0 \leq i < K$ korral on **P[i]** ühe väljapääsuga kambri number; kambri **0** väljapääsu ei ole.

Funktsioon peab tagastama minimaalse **T**, mille korral leidub hea põgenemisplaani.

Võib eeldada, et igast kambrist, mis pole väljapääs, väljub vähemalt kaks koridori. Lisaks võib eeldada, et igas testis leidub vähemalt üks hea põgenemisplaani, mille korral $T \leq 1\,000\,000\,000$.

Näited

Näide 1

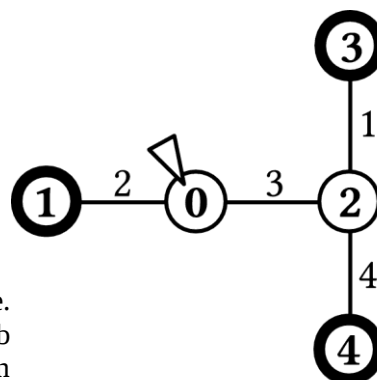
Vaatleme joonisel 1 toodud näidet, kus $N=5$, $M=4$, $K=3$,

$$R = \begin{pmatrix} 0 & 1 \\ 0 & 2 \\ 3 & 2 \\ 2 & 4 \end{pmatrix} \quad L = \begin{pmatrix} 2 \\ 3 \\ 1 \\ 4 \end{pmatrix} \quad P = \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix}$$

Ringid kujutavad kambreid, neid ühendavad jooned koridore. Väljapääsudega kambrid on näidatud paksu piirjoonega. Benjamas alustab liikumist kambrist 0 (tähistatud kolmnurgaga). Optimaalne plaan on järgmine:

- ⤴ Kui jõuad kambrisse 0, mine edasi kambrisse 1; kui see koridor on kinni, mine kambrisse 2.
- ⤴ Kui jõuad kambrisse 2, mine edasi kambrisse 3; kui see koridor on kinni, mine kambrisse 4.

Halvimal juhul jõuab Benjamas väljapääsuni 7 ajaühiku pärast. Seega peab `travel_plan` tagastama 7.



Joonis 1.

Näide 2

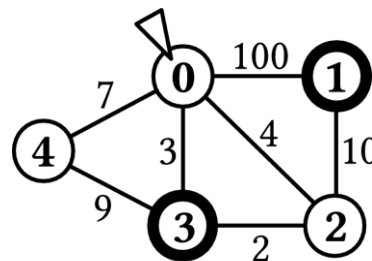
Vaatleme joonisel 1 toodud näidet, kus $N=5$, $M=7$, $K=2$,

$$R = \begin{pmatrix} 0 & 2 \\ 0 & 3 \\ 3 & 2 \\ 2 & 1 \\ 0 & 1 \\ 0 & 4 \\ 3 & 4 \end{pmatrix} \quad L = \begin{pmatrix} 4 \\ 3 \\ 2 \\ 10 \\ 100 \\ 7 \\ 9 \end{pmatrix} \quad P = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

Optimaalne plaan on järgmine:

- ⤴ Kui jõuad kambrisse 0, mine edasi kambrisse 3; kui see koridor on kinni, mine kambrisse 2.
- ⤴ Kui jõuad kambrisse 2, mine edasi kambrisse 3; kui see koridor on kinni, mine kambrisse 1.
- ⤴ Kamber 4 ei puutu asjasse, sinna ei jõua sa kunagi.

Benjamas jõuab väljapääsuni hiljemalt 14 ajaühiku pärast. Seega peab `travel_plan` tagastama 14.



Joonis 2.

Alamülesanded

Alamülesanne 1 (46 punkti)

- ✦ $3 \leq N \leq 1\,000$.
- ✦ Krokodillide linna struktuur on puu. See tähendab, et $M = N-1$ ja iga kambrite paari i ja j jaoks leidub koridore jada, mis neid ühendab.
- ✦ Igast väljapääsuga kambrist saab ainult ühte teise kambrisse.
- ✦ Iga muu kamber on vahetult ühendatud kolme või enama teise kambriga.

Alamülesanne 2 (43 punkti)

- ✦ $3 \leq N \leq 1\,000$.
- ✦ $2 \leq M \leq 100\,000$.

Alamülesanne 3 (11 punkti)

- ✦ $3 \leq N \leq 100\,000$.
- ✦ $2 \leq M \leq 1\,000\,000$.

Realisatsioon

Piirangud

- ✦ Protsessori aeg: 2 sekundit
- ✦ Mälu: 256 MB

Liides (API)

- ✦ Töökataloog: crocodile/
- ✦ Võistleja lahendus: crocodile.cpp
- ✦ Võistleja lahenduse liides: crocodile.h
- ✦ Testkeskkond: grader.cpp
- ✦ Testkeskkonna sisendi näited: grader.in.1, grader.in.2

Märkus: Testkeskkonna sisendi vorming:

- ✦ Rida 1: N, M, K .
- ✦ Read 2... $M+1$: real $i+2$ ($0 \leq i < M$) on tühikutega eraldatud $R[i][0], R[i][1], L[i]$.
- ✦ Rida $M+2$: K tühikutega eraldatud täisarvu: $P[0], P[1], \dots, P[K-1]$.
- ✦ Rida $M+3$: eeldatav vastus.
- ✦ Oma lahenduse testimiseks kompileerimine:
g++ -o crocodile crocodile.cpp grader.cpp
- ✦ Oma lahenduse testimiseks käivitamine:
cat grader.in.1 | ./crocodile