

Endurbyggja

Tímatakmörk: 4 sekúndur

Minnistakmörk: 1024 MiB

Þetta er gagnvirk verkefni.

Bissy er við það að taka þátt í leiknum *Loka áfangastaður* sem sjálfboðaliðarnir skipulögðu. Þar munu EJOI liðin keppa í að heimsækja mismunandi staðsetningar um borgina. En Bissy hefur meiri áhuga á því hvernig leikurinn var hannaður en að vinna hann.

Það eru N staðir á kortinu og N lið. Sérhvert lið verður að heimsækja alla staði. Sérhvert lið hefur sinn eigin upphafsstað, það er lið i byrjar frá stað i . Hvert liði fær sína eigin leið.

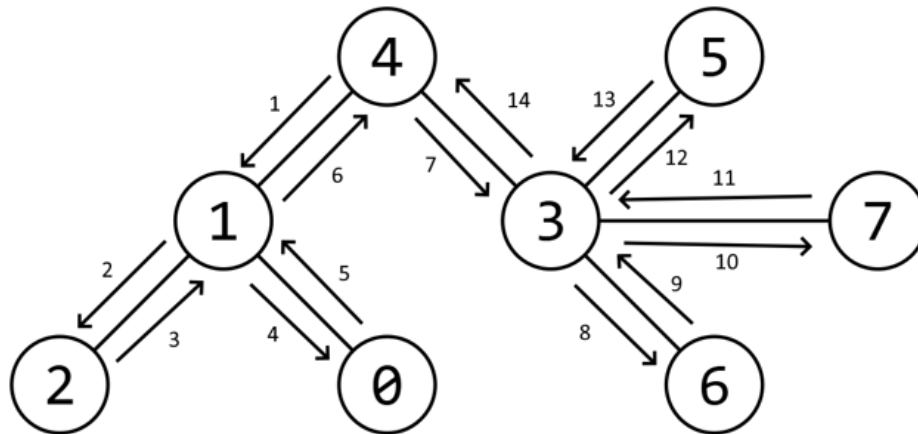
Bissy hefur einnig einhverjar innherjaupplýsingar. Leiðirnar eru byggðar á faldri uppbyggingu fljótra almenningssamgöngulína. Nánar tiltekið, keppnisnefndin valdi $N - 1$ línur, þar sem hver lína tengir par staða. Þau gera það þannig að frá hverjum stað er hægt að komast til hvaða annars staðar sem er með því að nota þessar línur. Slík uppbygging er einnig kölluð tré. Síðan, fyrir hvern upphafsstað i , mynduðu dómarar **handahófskenndan** DFS veg sem eru skilgreindur hér að neðan og úthlutuðu þessum vegi sem leið fyrir lið i .

Bissy vill komast að því hvernig almenningssamgöngutréð sem dómarar völdu er í laginu. Hún spyr aðeins hraðra spurninga af taginu:

- Hver er áfangastaður j fyrir lið i ?

Aðstoðaðu Bissy með því að skrifa forrit `find_tree` sem finnur falda tréð.

DFS vegur trés er röðin sem hnútar þess eru heimsóttir í fyrsta skipti af dýptarleit. Slíkt ferli byrjar á gefnum hnút v og fer endurkvæmt til einhvers af óheimsóttu nágrönnum hans þar til engir slíkir nágrannar eru til. Þá fer það aftur til áður heimsóttis hnúts og heldur áfram þaðan. Dæmi:



Hér er $v = 4$ og örvarnar meðfram leggjunum sýna skrefin í djúpleit. Framkallaði vegurinn er $[4, 1, 2, 0, 3, 6, 7, 5]$. Athugaðu að röðin sem við heimsækjum nágranna í skiptir máli þar sem annar vegur gæti verið $[4, 3, 5, 7, 6, 1, 0, 2]$.

Trjáuppbyggingin og N DFS vegirnir eru ákveðnir áður en forrit þitt hefst og breytast ekki þegar svarað er spurningum þínum. Raðir nágranna hnúta í þessum DFS geta breyst.

Útfærslusmáatriði

Þú verður að útfæra eftirfarandi fall:

```
std::vector<std::pair<int, int>> find_tree(int N)
```

- hér er N fjöldi staðsetninga;
- skilagildið er listi af leggjum trésins þar sem röð leggjana eða röð endapunkta á hverjum legg skipta ekki máli.

Fyrir hvert próf gæti verið beitt þessu falli allt að T sinnum.

Til að eiga samskipti við dómnefndina máttu beita eftirfarandi falli:

```
int guess(int i, int j)
```

Það skilar staðsetningu j á vegi liðs i í trénu. Athugaðu að `guess(i, 0)` mun skila i . Þú mátt gera ráð fyrir að fallið svari í fastri tímaflækju $O(1)$ fyrir öll hlutverkefni upp að hlutverkefni 6 og í logratíma $O(\log N)$ fyrir hlutverkefni 7. Það á alltaf að gilda að $0 \leq i, j \leq N - 1$, því annars mun lausn þín fá dóminn `Output isn't correct: Invalid call`.

Takmarkanir

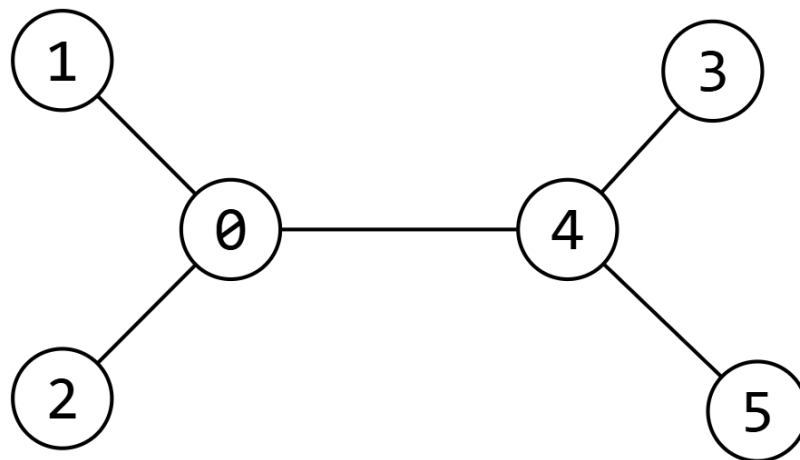
- $2 \leq N \leq 2^{16} + 1$
- T fylgir takmörkunum sem byggjast á stærstu gildum N (táknuð með N_{max}) meðal fallbeitinga í sérhverju prófu:

- ef $N_{max} \leq 9$, þá er $1 \leq T \leq 100$
- ef $N_{max} \leq 2^{10} + 1$, þá er $1 \leq T \leq 10$
- ef $N_{max} \leq 2^{16} + 1$, þá er $1 \leq T \leq 3$

Kerfisýfirferðarforritið getur notað allt að 280 MiB, sem telst til minnis lausnar þinnar.

Sýnidæmi

Assume the hidden structure of fast public transport lines is as follows:

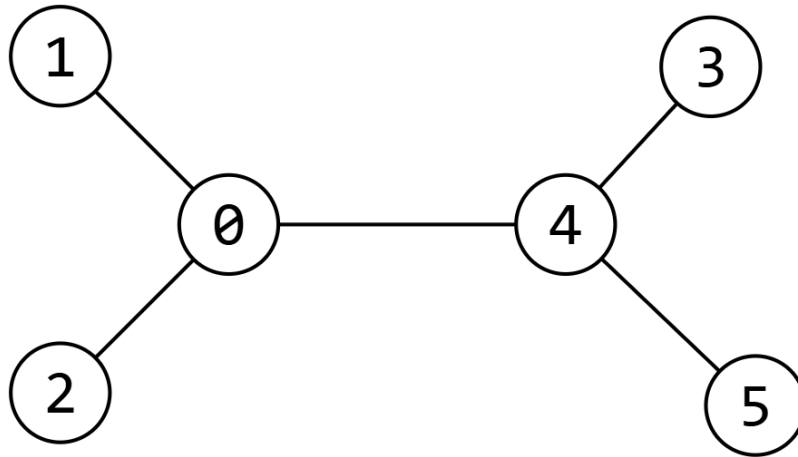


Suppose that for the starting location 0, the walk is $[0, 1, 2, 4, 3, 5]$. Then the following is an example interaction:

Participant program	Jury program
	<code>find_tree(6)</code>
<code>guess(0, 0)</code>	0
<code>guess(0, 1)</code>	1
<code>guess(0, 2)</code>	2
<code>guess(0, 3)</code>	4
<code>guess(0, 4)</code>	3
<code>guess(0, 5)</code>	5
<code>return {{0,1},{0,2},{4,0},{5,4},{3,4}};</code>	

Note that the information from the queries is not enough to uniquely determine the tree, but this is in fact the answer to the test in subtask 0.

Gerðu ráð fyrir að falda uppbyggingin á almenningssamgöngulínunum sé eftirfarandi:



Gerðu ráð fyrir að fyrir upphafsstað 0 sé gönguleiðin $[0, 1, 2, 4, 3, 5]$. Þá er eftirfarandi dæmi um samskipti:

Forrit þátttakanda	Forrit dómnefndar
	<code>find_tree(6)</code>
<code>guess(0, 0)</code>	0
<code>guess(0, 1)</code>	1
<code>guess(0, 2)</code>	2
<code>guess(0, 3)</code>	4
<code>guess(0, 4)</code>	3
<code>guess(0, 5)</code>	5
<code>return {{0,1},{0,2},{4,0},{5,4},{3,4}};</code>	

Athugaðu að upplýsingarnar úr fyrirspurnum eru ekki nægar til að ákvarða tréð með fullri vissu, en þetta er í raun svarið við prófinu í hlutverkefni 0.

Hlutverkefni

Hlutverkefni	Stig	N	Frekari takmarkanir
0	0	-	Sýnidæmið.
1	11	≤ 9	-
2	6	≤ 100	Sérhver hnútur er tengdur við mest 2 aðra hnúta.
3	13	≤ 100	Sérhver vegur er framkallaður með djúpleit sem forgangsraðar í hverju skrefi að færa sig frá hnúti 0. Ef það eru margir kostir til að færa sig frá hnúti 0 er einn þeirra valinn af handahófi.
4	11	≤ 100	Sérhver hnútur annar en hnútur 0 er tengdur við mest 2 aðra hnúta.
5	10	≤ 100	-
6	31	$\leq 2^{10} + 1$	-
7	18	$\leq 2^{16} + 1$	-

Stigagjöf

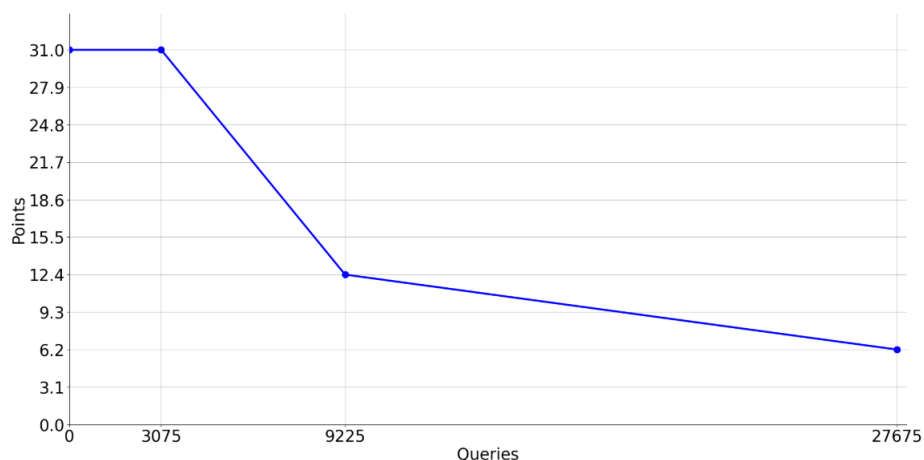
Fyrir hvert hlutverkefni upp að hlutverkefni 5 færðu fullt hús stiga ef þú getur fundið tréð innan tímamarkanna. Fyrir hlutverkefni 6 og 7 færðu brot S af fullu húsi stiga út frá **hámarksfjölda** spurninga sem þú spyrð á undirprófi, Q_{max} . Stigagjöfin fyrir þessi hlutverkefni er eftirfarandi:

- if $Q_{max} \leq L_1$, þá er $S = 1.0$;
- if $L_1 < Q_{max} \leq L_2$, þá er $S = 0.4 + 0.6 \times \frac{L_2 - Q_{max}}{L_2 - L_1}$;
- if $L_2 < Q_{max} \leq 3L_2$, þá er $S = 0.2 + 0.2 \times \frac{3L_2 - Q_{max}}{2L_2}$;
- if $3L_2 < Q_{max}$, þá er $S = 0.2$;

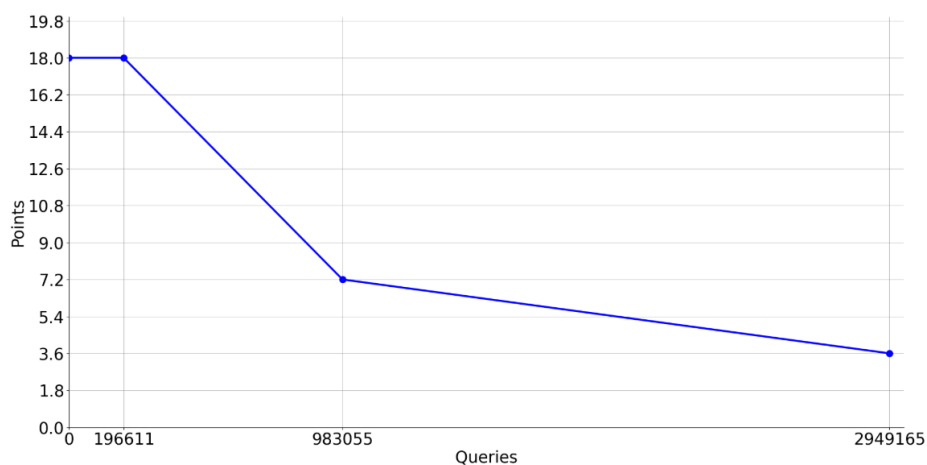
þar sem $L_1 = 3 \times (2^{10} + 1) = 3075$ og $L_2 = 9 \times (2^{10} + 1) = 9225$ fyrir hlutverkefni 6. Fyrir hlutverkefni 7 er $L_1 = 3 \times (2^{16} + 1) = 196611$ og $L_2 = 15 \times (2^{16} + 1) = 983055$. Brot stiga sem þú færð fyrir hlutverkefnið í heild sinni er minnsta S yfir öll próf.

Línuritinn að neðan sýna stigagjafarfallið fyrir síðustu tvö hlutverkefnin:

Hlutverkefni 6



Hlutverkefni 7



Sýnisyfirferðarforrit

Það eru tvö sýnisyfirferðarforrit gefin.

Fyrir staðbundna prófun á eigin vél má nota `Lgrader.cpp` saman með þýðingu á eigin forriti og þannig vera með tól til að prófa. Forritið les fjölda prufutilvika T Fyrir sérhvert prufutilvik býst tólið við fjölda staðsetninga N og svo $N - 1$ línur, sérhver þeirra með þar heiltalna sem lýsa beinni tengingu. Eftir það fylgja N línur, hver með N heiltölum, sem lýsa DFS vegunum. Athugaðu að vegur i verður að byrja frá hnút i . Þú getur látið yfirferðarforritið framkalla vegi með því að stilla `AUTO_GENERATE` fastann sem `true`. Sem úttak mun yfirferðarforritið annaðhvort tilkynna villuskilaboð ef niðurstaðan er röng eða fjölda fyrirspurna sem þú spurðir fyrir hvert prufutilvik og svo hámark yfir heildina. Athugaðu einnig að yfirferðarforritið virkar ekki fyrir nægilega stór gildi af N , nánar tiltekið fyrir takmarkanirnar í hlutverkefni 7, þar sem þú mátt beita ímyndunaraflinu í staðin fyrir það tilfelli.

Fyrir kerfisnotandaprófanir má notast við `stub.cpp` með eigin forriti fyrir notandaprófanir á kerfinu. Snið inntaksins er hið sama og í hinu yfirferðarforritinu. Athugaðu að fyrir

notandaprófanir er ekki innbyggð leið tli að framkalla vegin.