

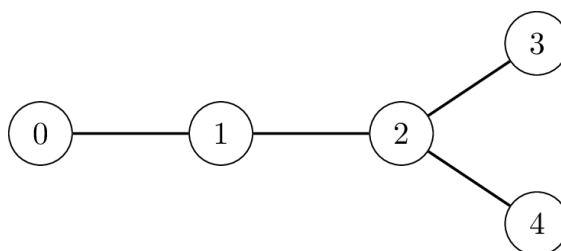
## Миграции

Природонаучният музей проучва миграционните модели на динозаври в Боливия. Палеонтолозите са открили отпечатьци от стъпки на динозаври на  $N$  различни обекта, номерирани от 0 до  $N - 1$  в намаляващ по възраст ред: в обект 0 се намират най-старите отпечатьци, докато в обект  $N - 1$  — най-младите.

Динозаврите са мигрирали към всеки обект (освен обект 0) от определен, по-стар обект. За всеки обект  $i$ ,  $1 \leq i \leq N - 1$ , съществува точно един по-стар обект  $P[i]$  (с  $P[i] < i$ ), за който някои динозаври са мигрирали директно от обект  $P[i]$  до обект  $i$ . Един по-стар обект може да е бил източник на миграция към няколко по-млади обекти.

Палеонтолозите моделират всяка миграция като **ненасочено ребро** между обекти  $i$  и  $P[i]$ . Обърнете внимание, че за два различни обекта  $x$  и  $y$ , винаги е възможно да се пътува от  $x$  до  $y$  чрез преминаване през последователност от ребра. **Разстоянието** между обекти  $x$  и  $y$  се дефинира като минималния брой ребра, нужни за пътуване от  $x$  до  $y$ .

Например, следната илюстрация показва случай с  $N = 5$  обекта, където  $P[1] = 0$ ,  $P[2] = 1$ ,  $P[3] = 2$ , и  $P[4] = 2$ . Може да се пътува от обект 3 до обект 4 чрез 2 ребра, така че разстоянието между тях е 2.



Музеят цели да идентифицира двойка обекти с максимално възможно разстояние. Забележете, че двойката не е задължително единствена: ако разгледаме примера по-горе, двете двойки  $(0, 3)$  и  $(0, 4)$  имат разстояние 3, което е максимумът. В такива случаи, всяка валидна двойка, с максимално разстояние, е считана за валидна.

В началото стойностите  $P[i]$  са **неизвестни**. Музеят изпраща изследователски екип да посети обектите  $1, 2, \dots, N - 1$ , в този ред. При достигане на обект  $i$  ( $1 \leq i \leq N - 1$ ), екипът извършва следните две действия:

- Разбира стойността  $P[i]$ , тоест източника на миграция към обект  $i$ .
- Решава дали да изпрати **едно** съобщение към музея или да не изпрати съобщение от този обект, в зависимост от досега събраната информация.

Съобщенията биват предавани чрез скъпа сателитна комуникационна система и затова всяко съобщение трябва да е цяло число между 1 и 20 000, включително. Освен това, екипът може да изпраща общо **най-много 50 съобщения**.

Вашата задача е да имплементирате стратегия, чрез която:

- Изследователският екип избира обектите, от които изпраща съобщения, заедно със стойността за всяко съобщение.
- Музеят може да определи двойка обекти с максимално разстояние, използвайки само съобщенията получени от всеки обект (и знаейки от кои обекти са изпратени съобщенията).

Изпращането на големи числа чрез сателита е скъпо. Вашият резултат ще зависи от най-голямото изпратено число и общия брой изпратени съобщения.

## Детайли по имплементацията

Трябва да имплементирате две функции; една за изследователския екип и една за музея.

Функцията за **изследователския екип** е:

```
int send_message(int N, int i, int Pi)
```

- $N$ : броят обекти със следи от динозаври.
- $i$ : индексът на обекта, който екипът изследва в момента.
- $Pi$ : стойността  $P[i]$ .
- Функцията бива извикана  $N - 1$  пъти за всеки тест, в реда  $i = 1, 2, \dots, N - 1$ .

Функцията трябва да връща  $S[i]$ , което определя действието, взето от изследователския екип на обект  $i$ :

- $S[i] = 0$ : изследователският екип решава да не изпраща съобщение от обект  $i$ .
- $1 \leq S[i] \leq 20\,000$ : изследователският екип изпраща  $S[i]$  като съобщението от обект  $i$ .

Функцията за **музея** е:

```
std::pair<int,int> longest_path(std::vector<int> S)
```

- $S$ : масив с дължина  $N$ , за който:
  - $S[0] = 0$ .

- За всяко  $1 \leq i \leq N - 1$ ,  $S[i]$  е цялото число върнато от `send_message(N, i, P[i])`.
- Функцията бива извикана точно веднъж за всеки тест.

Функцията трябва да върне двойка обекти  $(U, V)$  с максимално разстояние.

В истинското оценяване, програмата, която извиква горните функции, бива изпълнена **точно два пъти**.

- През първото изпълнение на програмата:
  - `send_message` бива извикана  $N - 1$  пъти.
  - **Вашата програма може да съхранява и запазва информация и да я използва в последващи извиквания на функцията.**
  - Върнатите стойности (масив  $S$ ) биват съхранени от системата.
  - В някои случаи грейдърът е **адаптивен**. Това означава, че стойностите  $P[i]$  в извикване на `send_message` могат да зависят от предишни действия на изследователския екип.
- През второто изпълнение на програмата:
  - `longest_path` бива извикан точно веднъж. Единствената информация достъпна до `longest_path` от първото изпълнение е масива  $S$ .

## Ограничения

- $N = 10\,000$
- $0 \leq P[i] < i$  за всяко  $i$ , за което  $1 \leq i \leq N - 1$ .

## Подзадачи и оценяване

| Подзадача | Точки | Допълнителни ограничения                                                             |
|-----------|-------|--------------------------------------------------------------------------------------|
| 1         | 30    | Обект 0 и някой друг обект са на максимално разстояние измежду всички двойки обекти. |
| 2         | 70    | Няма допълнителни ограничения.                                                       |

Нека  $Z$  е максималното число в  $S$  и нека  $M$  е броят съобщения, изпратени от изследователския екип.

Във всеки от тестовете, ако поне едно от следните условия е в сила, резултатът на Вашето решение за този тест ще бъде 0 (показано като `Output isn't correct` в CMS):

- Поне един елемент в  $S$  е невалиден.
- $Z > 20\,000$  или  $M > 50$ .
- Върната стойност при извикване на функцията `longest_path` е грешна.

Иначе, Вашият резултат за подзадача 1 се изчислява по следния начин:

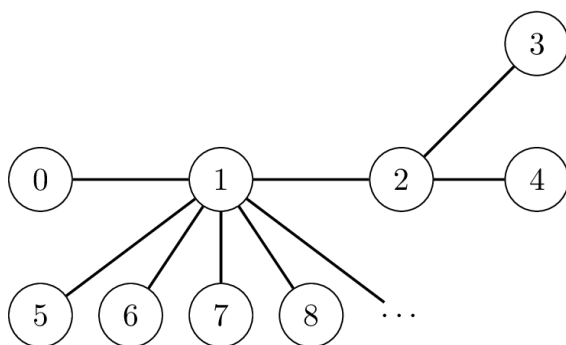
| Условие                      | Точки |
|------------------------------|-------|
| $9\,998 \leq Z \leq 20\,000$ | 10    |
| $102 \leq Z \leq 9997$       | 16    |
| $5 \leq Z \leq 101$          | 23    |
| $Z \leq 4$                   | 30    |

Вашият резултат за подзадача 2 се изчислява по следния начин:

| Условие                               | Точки                                            |
|---------------------------------------|--------------------------------------------------|
| $5 \leq Z \leq 20\,000$ и $M \leq 50$ | $35 - 25 \log_{4000} \left( \frac{Z}{5} \right)$ |
| $Z \leq 4$ и $32 \leq M \leq 50$      | 40                                               |
| $Z \leq 4$ и $9 \leq M \leq 31$       | $70 - 30 \log_4 \left( \frac{M}{8} \right)$      |
| $Z \leq 4$ и $M \leq 8$               | 70                                               |

## Пример

Нека  $N = 10\,000$ . Да разгледаме ситуация, при която  $P[1] = 0$ ,  $P[2] = 1$ ,  $P[3] = 2$ ,  $P[4] = 2$ , и  $P[i] = 1$  за  $i > 4$ .



Приемете, че стратегията на изследователите е да изпратят съобщение  $10 \cdot V + U$  когато двойката места  $(U, V)$  с максимално разстояние се променя като резултат от извикване на `send_message`.

В началото, двойката с максимално разстояние е  $(U, V) = (0, 0)$ . Нека разгледаме следната последователност от извиквания в първото изпълнение на програмата.

| Извикване на функция                   | $(U, V)$ | Върната стойност ( $S[i]$ ) |
|----------------------------------------|----------|-----------------------------|
| <code>send_message(10000, 1, 0)</code> | $(0, 1)$ | 10                          |
| <code>send_message(10000, 2, 1)</code> | $(0, 2)$ | 20                          |
| <code>send_message(10000, 3, 2)</code> | $(0, 3)$ | 30                          |
| <code>send_message(10000, 4, 2)</code> | $(0, 3)$ | 0                           |

Забележете, че във всички останали извиквания, стойността на  $P[i]$  е 1. Това означава, че двойката с максимално разстояние не се променя, и че изследователският екип не изпраща повече съобщения.

Тогава във второто изпълнение на програмата, следното извикване бива извършено:

```
longest_path([0, 10, 20, 30, 0, ...])
```

Музеят чете последното съобщение изпратено от изследователския екип, което е  $S[3] = 30$ , и заключава, че  $(0, 3)$  е двойката обекти с максимално разстояние. Следователно, функцията връща  $(0, 3)$ .

Забележете, че този подход не винаги позволява на музея да заключи правилно двойката с максимално разстояние.

## Локален грейдър

Локалният грейдър извиква `send_message` и `longest_path` в едно и също изпълнение, което е различно поведение от това на истинския грейдър.

Формат на входа:

```
N
P[1] P[2] ... P[N-1]
```

Формат на изхода:

```
S[1] S[2] ... S[N-1]
U V
```

Забележете, че можете да използвате локалния грейдър с произволна стойност на  $N$ .