

SEARCH

Lễ cưới của RR sắp đến gần mà các vị khách vẫn chưa tìm được biệt thự của RRHY, nằm trên phố X huyền thoại. Không có thời gian điều phối, RR đành nhờ bạn chỉ đường cho các vị khách. Lập tức bạn chạy vì vu ra đón các vị khách quý. Không may, chính bạn cũng không nhớ đường về! Nhưng tính cả rồi, bạn đã có được máy dò đường 3000, một thiết kế tinh tế của bạn và RR.

Khu phố X gồm có N biệt thự đánh số từ 1 đến N , được nối với nhau bởi $N - 1$ con đường hai chiều. Từ một biệt thự bất kỳ có thể sang một biệt thự bất kỳ khác bằng các con đường nối. Biệt thự của RR được đánh số P . Bạn và các vị khách thì đang ở biệt thự số Q .

Máy dò đường của RR được cài sẵn một dãy số $A[i]$, hiện ra tương ứng với biệt thự thứ i . Tuy nhiên, khi bạn đang ở biệt thự Q , máy dò đường chỉ cho bạn biết chỉ số x và $A[x]$ của các biệt thự kề với Q , cũng như chỉ số Q và $A[Q]$. Bạn cần tìm hướng đi ngắn nhất để trở về biệt thự của RR, hãy thiết kế chiếc máy dò đường để hoạt động như mong muốn!

Do bộ nhớ có hạn, nên các phần tử của mảng $A[]$ đều phải là các số nguyên trong khoảng từ 0 đến V .

Đặc tả cài đặt

Lưu ý: Bạn cần bổ sung thư viện `search.h` khi cài đặt. Máy dò đường phải được cài đặt trước bởi hàm sau:

```
vector<int> install(int N, vector<int> u, vector<int> v, int P, int K);
```

Trong đó:

- N là số biệt thự trong khu phố.
- u và v là hai mảng có $N - 1$ phần tử, $u[i]$ và $v[i]$ thể hiện con đường thứ i trong đường đi.
- P là chỉ số biệt thự của RR.
- K là chỉ số của bộ dữ liệu bạn đang giải.

Hàm sẽ trả về một mảng N phần tử là mảng $A[]$, trong đó $A[i]$ sẽ là chỉ số được gán cho biệt thự i .

Bạn phải tìm đường theo thiết kế, bằng hàm sau:

```
int find_way(int Q, int aQ, vector<int> Adj, vector<int> aAdj, int K);
```

Trong đó:

- Q là biệt thự hiện tại bạn đang đứng.
- aQ là giá trị $A[Q]$.
- Adj và $aAdj$ là hai mảng có độ lớn bằng nhau, trong đó $Adj[i]$ là chỉ số đỉnh kề thứ i của Q , và $aAdj[i]$ là $A[Adj[i]]$.
- K là chỉ số của bộ dữ liệu bạn đang giải.

Bạn phải trả về:

- Q nếu Q chính là biệt thự của RR.
-

Happy Wedding Contest - Round 2

- Chỉ số biệt thự tiếp theo phải đi trên đường đi ngắn nhất đến biệt thự của RR, nếu ngược lại.

Bạn có thể tìm thấy cài đặt cơ bản của hai hàm này trong tệp `search.cpp`.

Trình chấm và biên dịch trình chấm

Trình chấm sẽ gọi lần lượt hai hàm `install` và `find_way` theo đúng thứ tự.

Bạn có thể tìm thấy bản cài đặt mẫu của trình chấm trong tệp `grader.cpp`.

Để biên dịch trình chấm, sử dụng lệnh:

`g++ -O2 -o search search.cpp grader.cpp` trên Windows/Linux, hoặc chạy tệp `compile.sh` trên Linux. Trình chấm sẽ đọc dữ liệu theo định dạng sau, từ bàn phím:

- Dòng đầu tiên gồm bốn số N, K, P và Q là số biệt thự, chỉ số của bộ dữ liệu bạn đang giải, biệt thự của RR và biệt thự bạn đang đứng.
- $N - 1$ dòng tiếp theo, mỗi dòng gồm hai số u và v miêu tả một đường đi nối giữa biệt thự u và v .

Trình chấm sẽ xuất ra màn hình theo định dạng:

- Một dòng duy nhất ghi đỉnh tiếp theo, theo kết quả của hàm `find_way`, hoặc
- `Incorrect`, nếu các hàm của bạn không xuất ra định dạng thỏa mãn.

Chấm điểm

Bạn sẽ được toàn bộ điểm của test nếu trả lời đúng, 0 điểm nếu trả lời sai.

Giới hạn

- $2 \leq N \leq 100000$.
- $1 \leq P, Q \leq N$.
- $1 \leq u, v \leq N$.
- $1 \leq K \leq 4$.

Bộ dữ liệu

- Bộ dữ liệu 1 (10% số điểm của bài): $K = 1, V = N - 1$.
 - Bộ dữ liệu 2 (15% số điểm của bài): $K = 2, V = 2$.
 - Bộ dữ liệu 3 (20% số điểm của bài): $K = 3, V = 1$, không có biệt thự nào có quá hai con đường nối chúng, P khác Q .
 - Bộ dữ liệu 4 (55% số điểm của bài): $K = 4, V = 1$.
-