

Los Binaires

locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Los Binaires е най-модерният град построен през 2020 в който няма имена на улици, защото е твърде трудно да се помнят, а всеки адрес е едно единствено число. За да е по-лесно навигирането от всеки блок може да се отиде директно до най-много 2 други. За всеки един блок Q: ако има блокове с адреси по-малки от адреса на Q - те са на ляво от Q, а ако има такива с адреси по-големи от Q - те са на дясно. Тъй като всеки блок има директна връзка най-много с 2 други, няма други възможности. Не е възможно да има блокове с еднакви адреси.

Най-големият онлайн магазин Маазон иска да прави доставки в този град и иска да му напишете алгоритъм, който да изчислява колко блока трябва да посети, за да достигне целта.

На входа ще получите последователността от адреси, които са използвани за построяването на града, като ако някой се опита да построи блок с адрес, който вече съществува не трябва да се случва нищо, защото всички блокове имат уникални адреси. След което ще получите заявките от Маазон.

Input Format

N K - ще получите последователност от N адреса по които да построите града, след което ще получите K заявки от Маазон колко е отдалечен даден адрес от началният, където е офиса на Маазон

B_1 B_2 ... B_N - адресите по които трябва да построите града (ако 2 адреса се повтарят е валиден само първият)

R_1 R_2 ... R_K - адресите на които Маазон иска да доставя, за които трябва да определите на какво разстояние са от първата построена сграда. Първата построена сграда е на разстояние 0 от себе си :)

Constraints

$$1 \leq N \leq 10^5$$

$$1 \leq K \leq 10^5$$

$$INT_MIN \leq B_i \leq INT_MAX$$

$$INT_MIN \leq R_i \leq INT_MAX$$

Височината на "града" няма да счупи нормален алгоритъм за решаването на задачата, тестовете са рандомизирани.

Output Format

K - на брой реда, като на ред i е разстоянието от офиса на Маазон до адреса от заявка R_i . Ако не съществува такъв адрес разстоянието е -1

Sample Input 0

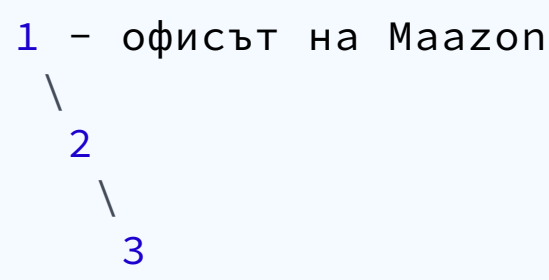
```
3 2
1 2 3
2 4
```

Sample Output 0

```
1
-1
```

Explanation 0

Градът ще изглежда по следния начин



Следователно 1 е на разстояние 0. 2 е на разстояние 1. 3 е на разстояние 2. Съответно адреси, които не съществуват като 4 ще са -1.

Sample Input 1

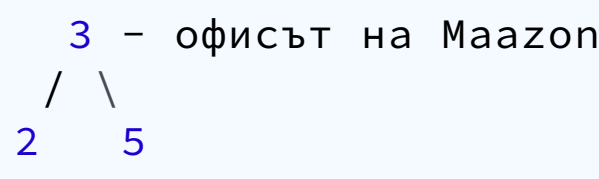
```
3 3
3 5 2
3 5 0
```

Sample Output 1

```
0
1
-1
```

Explanation 1

Градът ще изглежда по следния начин



Следователно 3 е на разстояние 0. 2 е на разстояние 1. 5 е на разстояние 1. Съответно адреси, които не съществуват като 0 ще са -1.

Current Buffer (saved locally, editable)

C++

1

#include <cmath>

2

#include <cstdio>

3

#include <vector>

4

#include <iostream>

5

#include <algorithm>

6

int counter=-1;

7

using namespace std;

8

struct Node{

9

int data;

10

Node* left;

11

Node* right;

12

Node(int value){

13

data=value;

14

}

15

};

16

class BST{

17

public:

18

Node* root=nullptr;

19

public:

20

Node* _insert(int value,Node* current){

21

22

if(current==nullptr){

23

return new Node(value);

24

}

25

else if(current->data<value){

26

current->right=_insert(value,current->right);

27

}

28

else if(current->data>value){

29

current->left=_insert(value,current->left);

30

}

31

return current;

32

}

33

void insert(int value){

34

root=_insert(value,root);

35

}

36

bool _search(int value,Node* current){

37

counter++;

38

if(current){

39

if(current->data==value){

40

return true;

41

}

42

else if(current->data>value){

43

return _search(value,current->left);

44

}

45

else{

46

return _search(value,current->right);

47

}

48

}

49

else return false;

50

}

51

52

53

};

54

55

int main() {

56

BST tree;

57

int addressNumber;

58

int address;

59

int deliveryNumber;

60

int delivery;

61

cin>>addressNumber;

62

cin>>deliveryNumber;

63

for(int i=0;i<addressNumber;i++){

64

cin>>address;

65

tree.insert(address);

66

}

67

for(int i=0;i<deliveryNumber;i++){

68

cin>>delivery;

69

if(tree._search(delivery,tree.root)){

70

cout<<counter;

71

}

72

else{

73

cout<<-1;

74

}

75

counter=-1;

76

cout<<endl;

77

}

78

return 0;

79

}

80

Line: 1 Col: 1