

ساختمان داده دُری

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

دُری یک لیست از اعداد در اختیار دارد. او همیشه از بین این اعداد یک عنصر موردعلاقه دارد و عنصر موردعلاقه دُری همواره عددی است که اندیس $\lfloor \frac{n \cdot k}{100} \rfloor$ را دارد. n در هر لحظه برابر با تعداد اعداد لیست می‌باشد. (اندیس اعداد از صفر شروع می‌شود.) او می‌خواهد تعدادی عملیات را بر روی این لیست اعمال کند. عملیاتی‌هایی که دُری می‌تواند انجام دهد به شرح زیر است:

۱. یک عدد را به ابتدای لیست اضافه کند.

۲. یک عدد را به انتهای لیست اضافه کند.

۳. یک عدد را قبل از عنصر موردعلاقه اضافه کند.

۴. یک عدد را بعد از عنصر موردعلاقه اضافه کند.

۵. همه اعداد قبل از عنصر موردعلاقه را حذف کند.

۶. همه اعداد بعد از عنصر موردعلاقه را حذف کند.

(دقت کنید که بعد از هر عملیات طول لیست تغییر می‌کند و عنصر موردعلاقه ممکن است تغییر کند.)

دُری مطمئن نیست که عملیات‌ها را به درستی اجرا می‌کند برای همین از شما می‌خواهد لیست نهایی را چاپ کنید.

ورودی

در خط اول ورودی سه عدد n و k و q آمده است که با فاصله از یکدیگر جدا شده‌اند. n برابر با طول اولیه لیست، k برابر با متغیر تأثیرگذار در عنصر موردعلاقه و q برابر با تعداد عملیات‌ها است.

$$1 \leq n \leq 10^6$$

$$0 \leq k \leq 99$$

$$1 \leq q \leq 10^6$$

در خط بعد، n عدد با فاصله از یکدیگر آمده‌اند که a_i برابر با عدد i ام لیست اولیه است.

$$0 \leq i \leq n - 1$$

$$-10^8 \leq a_i \leq 10^8$$

در q خط بعد در هر خط یک عملیات آمده است. عملیات‌های 1 تا 4 به فرم زیر هستند.

- op X

که op شمارهٔ عملیات و x عددی است که به لیست اضافه می‌شود.

عملیات‌های 5 و 6 نیز به فرم زیر هستند.

- op

که op شمارهٔ عملیات است.

خروجی

در خط اول ابتدا تعداد اعداد لیست نهایی و در خط دوم همهٔ اعداد را به ترتیب با فاصله چاپ کنید.

مثال

ورودی نمونه ۱

```
5 50 2
1 2 3 4 5
4 7
```

5

خروجی نمونه ۱

3

7 4 5

ورودی نمونه ۲

2 5 4

7 3

3 1

1 5

2 4

6

خروجی نمونه ۲

1

5

ورودی نمونه ۳

6 67 9
-73500 -80831 -84276 -88522 -70642 -73038
3 5705
6
4 9961
6
3 4827
1 32391
1 3902
4 292
5

خروجی نمونه ۳

3
292 -88522 9961

ورودی نمونه ۴

3 0 2
3 4 5
3 2
1 1

خروجی نمونه ۴

5
1 2 3 4 5

ورودی نمونه ۵

105 74 81
-87948 -72650 -98850 -83059 -78276 -86034 -96570 -68893 -69809 -81993 -88663 -

2 27157

1 23622

5

3 24179

5

3 6191

4 22888

5

3 2634

3 20055

1 22646

5

6

2 20142

1 1416

6

3 18651

2 5699

6

5

6

3 2510

4 26869

6

4 9789

4 16423

5

3 10285

5

4 23757

5

6

6

3 31329

5

6

3 16549

2 9512

2 18060

5

6
2 25996
2 12529
2 17437
1 12949
2 23195
4 20416
3 16105
3 16282
6
1 11701
3 20671
3 12263
6
4 20053
1 10808
3 20945
6
2 19558
1 27982
2 4144
1 20222
2 2161
4 20450
2 10466
3 21659
1 26439
4 20024
1 29510
6
5
5
3 14018
2 9905
1 7391
3 3625
6
3 25824
1 25874
1 20159

2 28070

خروجی نمونه ۵

9

20159 25874 7391 21659 14018 25824 3625 20450 28070

ددلاین‌های امتحان

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

دُری و جواد مسئول برگزاری امتحان پایان‌ترم ساختمان‌های داده شده‌اند. در هر دقیقه تعدادی دانشجو وارد جلسه امتحان می‌شوند. دُری با توجه به شناختی که از بچه‌ها دارد، پس از ورود هر کس به جلسه به او t_i دقیقه وقت می‌دهد که پس از اتمام این t_i دقیقه، باید جلسه امتحان را ترک کند. از آنجا که جواد معتقد است جو امتحان باید هیجان‌انگیز باشد؛ در پایان هر دقیقه فردی را که به ترتیب حروف الفبایی کوچک‌ترین اسم را دارد، از جلسه بیرون می‌کند. جواد این کار را پس از اضافه شدن افراد جدید به جلسه و خروج افرادی که ددلاین آن‌ها پایان یافته است، انجام می‌دهد. اگر تعداد افراد حاضر در جلسه در آن دقیقه صفر نفر بود جواد در آن دقیقه هیچ کاری انجام نمی‌دهد.

حال از شما می‌خواهیم در n دقیقه ابتدایی از امتحان ورود افراد به جلسه را بررسی و نام افرادی که در هر دقیقه باید از جلسه خارج شوند (هم افرادی که وقت امتحان آن‌ها به پایان رسیده و هم فردی که جواد او را از جلسه بیرون می‌کند) را چاپ کنید.

ورودی

در خط اول ورودی، عدد n آمده است که برابر با تعداد دقیقه‌هایی است که باید بررسی شود.

$$1 \leq n \leq 10^5$$

در n خطی بعدی، در هر خط ابتدا عدد k آمده است که برابر با تعداد افرادی است که در آن دقیقه وارد جلسه می‌شوند و سپس به تعداد k رشته و عدد، به صورت $s_1 t_1 s_2 t_2 \dots s_k t_k$ با فاصله از یکدیگر آمده است که s_i برابر با نام فرد i ام و t_i برابر با زمانی است که دُری برای امتحان به آن فرد اختصاص می‌دهد.

$$1 \leq t_i$$

تضمین می‌شود که مجموع k ها و مجموع طول رشته‌ها از 10^5 کمتر است. تضمین می‌شود که تمامی اسم‌ها متفاوتند.

خروجی

خروجی برنامه شما باید شامل n خط باشد که در خط i ام ابتدا تعداد افرادی که باید در دقیقه i از جلسه خارج شوند و سپس اسامی آن‌ها را با فاصله به ترتیب صعودی چاپ کنید.

مثال

ورودی نمونه ۱

```
4
2 Amir 9 Pouya 2
1 Kian 10
1 Saman 5
0
```

خروجی نمونه ۱

```
1 Amir
1 Kian
2 Pouya Saman
0
```

ورودی نمونه ۲

```
7
1 Ali 4
2 Mehdi 8 Taha 4
1 MohammadReza 3
7 Zahra 6 Fatemeh 3 Reza 9 Sepehr 3 Morteza 2 Ahmad 1 Koosha 1
6 Parsa 2 Mehran 2 MohammadHossein 10 Mojtaba 2 Hossein 1 Hamed 1
```

3 Mobina 1 Arman 2 Dori 11
2 Attila 5 Javad 3

خروجی نمونه ۲

1 Ali
1 Mehdi
1 MohammadReza
1 Ahmad
2 Fatemeh Koosha
5 Arman Hamed Hossein Morteza Taha
6 Attila Mehran Mobina Mojtaba Parsa Sepehr

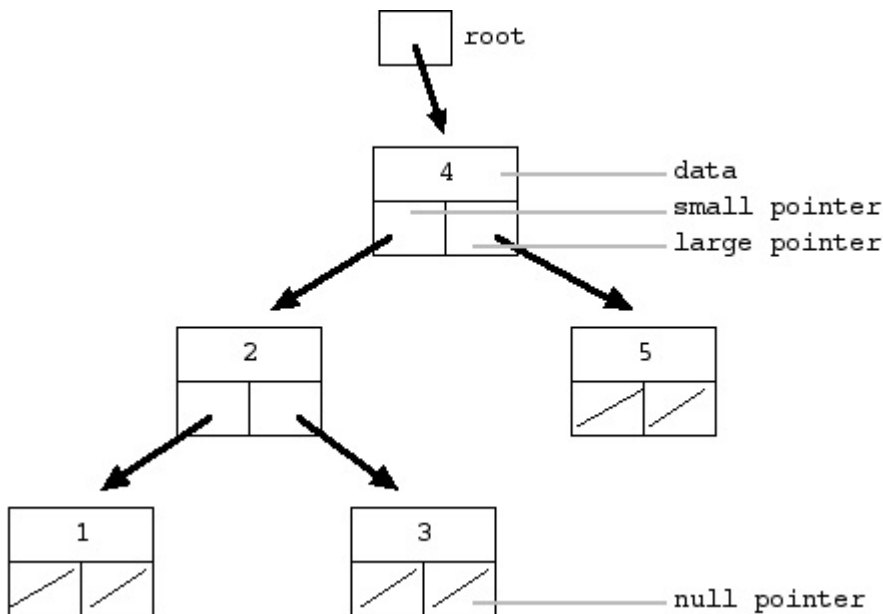
درخت پویای جواد

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

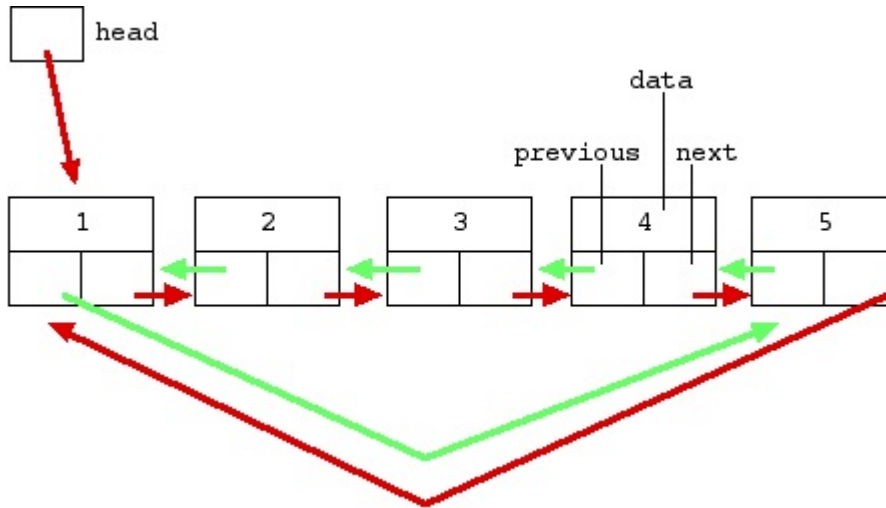
جواد بعد از اینکه فهمید دُری نمره خوبی توی درس ساختمان‌های داده گرفته، به این فکر افتاد تا یک سؤالی بهش بده که دُری نتونه حل کنه و غرورش بشکنه. جواد بعد از کلی فکرکردن تصمیم گرفت این سؤال رو به دُری بده:

یک درخت جست‌وجوی دودویی (BST) در اختیار شما قرار داده شده است. (BST درختی است که در آن مقدار هر گره از تمامی مقادیر گره‌های زیردرخت سمت راست آن کوچک‌تر بوده و از تمامی مقادیر گره‌های زیردرخت سمت چپ آن بزرگ‌تر است). می‌دانیم که هر گره یک اشاره‌گر به بچه سمت راست و یک اشاره‌گر به بچه سمت چپ خود دارد.



در مرحله اول به شما ریشه یک درخت با n گره داده می‌شود. شما وظیفه دارید که این درخت را به گونه‌ای تغییر دهید تا درخت شما به یک لیست پیوندی دوطرفه چرخشی مرتب‌شده تبدیل گردد به این معنی که اشاره‌گر سمت راست هر گره به کوچک‌ترین گره بزرگ‌تر از خود و اشاره‌گر سمت چپ هر گره به بزرگ‌ترین

گره کوچک‌تر از خود اشاره کند. (توجه کنید که اشاره‌گر سمت چپ کوچک‌ترین گره این درخت باید به بزرگ‌ترین گره و اشاره‌گر سمت راست بزرگ‌ترین گره این درخت باید به کوچک‌ترین گره اشاره کند.)



در مرحله دوم از شما به تعداد q سؤال پرسیده می‌شود. در هر سؤال اشاره‌گر به یک گره داده می‌شود و شما باید مجموع مقدار گره داده شده با مقدار گره قبل و بعد از آن در لیست پیوندی بدست آمده را برگردانید.

دو کد C++ و C# زیر در اختیار شما قرار داده شده است. شما می‌توانید هر یک را انتخاب کرده و آن‌ها را تغییر دهید به گونه‌ای که دو خواسته مسئله را برآورده کند. (دقت کنید که در صورت هر گونه تغییر در بخش‌های مشخص‌شده این کدها، نمره این سؤال را دریافت نخواهید کرد.)

```

1  #include <iostream>
2  using namespace std;
3
4  struct node
5  {
6      int value;
7      node *L, *R;
8  };
9
10 void process(node *c);
11
12 int query(node *x);
13
14 node *read_tree()
15
```

```
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    int n;
    cin >> n;
    node *A = new node[n];
    for (int i = 0; i < n; i++)
    {
        int v, li, ri;
        cin >> v >> li >> ri;
        A[i].value = v;
        if (li > 0)
            A[i].L = A + li;
        if (ri > 0)
            A[i].R = A + ri;
    }
    return A;
}

int main()
{
    node *tree = read_tree();
    process(tree);
    int q;
    cin >> q;
    while (q--)
    {
        int idx;
        cin >> idx;
        cout << query(tree + idx) << "\n";
    }
    return 0;
}

// =====DO NOT CHANGE ABOVE=====
// =====YOUR CODE BEGINS HERE=====

void process(node *c)
{
    // your code here
}
```

```
}

int query(node *x)
{
    // your code here
}
```

```
1  using System;
2  using System.Text;
3
4  internal class Program
5  {
6      class Node
7      {
8          public int value;
9          public Node Left;
10         public Node Right;
11         public Node() { }
12     }
13     static Node[] read_tree()
14     {
15         int n = int.Parse(Console.ReadLine());
16         Node[] nodes = new Node[n];
17         for(int i = 0; i < nodes.Length; i++)
18             nodes[i] = new Node();
19         for(int i = 0; i < n; i++)
20         {
21             var buffer = Console.ReadLine().Split().Select(int.Parse).ToArray();
22             int value = buffer[0];
23             int leftChild = buffer[1];
24             int rightChild = buffer[2];
25             nodes[i].value = value;
26             nodes[i].Left = leftChild > 0 ? nodes[leftChild] : null;
27             nodes[i].Right = rightChild > 0 ? nodes[rightChild] : null;
28         }
29         return nodes;
30     }
31
32     static void Main(string[] args)
33     {
34     }
```

```
{
    Node[] tree = read_tree();
    proccess(tree[0]);
    int queryCount = int.Parse(Console.ReadLine());
    StringBuilder stringBuilder = new StringBuilder();
    while (queryCount != 0)
    {
        int index = int.Parse(Console.ReadLine());
        stringBuilder.AppendLine(query(tree[index]).ToString());
        queryCount--;
    }
    Console.WriteLine(stringBuilder.ToString());
}

// =====DO NOT CHANGE ABOVE=====
// =====YOUR CODE BEGINS HERE=====

static void proccess(Node root)
{
    // your code here
}

static int query(Node node)
{
    // your code here
}
}
```

تابع *process* ریشه درخت را به عنوان ورودی دریافت می‌کند و درخت را به لیست پیوندی مرتب‌شده تبدیل می‌کند.

تابع *query* مقدار هر پرسش را محاسبه کرده و برمی‌گرداند.

به دُرّی کمک کنید تا بتونه سؤال جواد رو حل کنه.

ورودی

این بخش برای شما پیاده‌سازی شده است.

$$1 \leq n, q \leq 10^6$$

خروجی

این بخش برای شما پیاده‌سازی شده است.

مثال

ورودی نمونه ۱

```
5
4 2 3
3 -1 -1
2 4 1
5 -1 -1
1 -1 -1
3
0
3
2
```

خروجی نمونه ۱

```
12
10
6
```

این نمونه، درخت داده شده در شکل است. پرسش‌ها به ترتیب از راس‌های با مقادیر ۴، ۵ و ۲ هستند.