

# Финал - Бэкенд-разработка

🕒 1 июн 2019, 16:00:02

старт: 1 июн 2019, 12:00:00

финиш: 1 июн 2019, 16:00:00

длительность: 04:00:00

начало: 1 июн 2019, 12:00:00

конец: 1 июн 2019, 16:00:00

## В. Таргетинг на погоду

Ограничение времени	2 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Лето уже в самом разгаре — а это значит, что нас всех ждут и дождливые, и жаркие дни. В дождь людям нужны зонты и дождевики, и производителям этих товаров крайне важно отслеживать рост интереса к ним. С другой стороны, производителям прохладительных напитков важны жаркие дни, когда на продвижение можно потратить большие бюджеты. Итак, владельцам рекламных объявлений определённых тематик требуется возможность управлять настройками показов в зависимости от погоды. В этой задаче вам потребуется реализовать простейшую версию этой функциональности — таргетинг на температуру в определённые диапазоны дней.

В вашем распоряжении есть исторические данные о температуре, а также прогноз на ближайшее будущее. А именно:

- температура сегодня (этот день имеет номер  $0$ );
- температура за прошедшие  $N$  дней (вчерашний имеет номер  $-1$ , позавчерашний —  $-2$  и так далее до дня  $-N$ , самого раннего из известных);
- прогноз температуры на ближайшие  $N$  дней (завтрашний имеет номер  $1$ , послезавтрашний —  $2$  и так далее до дня  $N$ ).

Кроме того, есть данные о настройках показа  $B$  рекламных объявлений. Каждому из них соответствуют четыре числа  $t_{min}$ ,  $t_{max}$ ,  $d_{min}$ ,  $d_{max}$ , означающие, что объявление можно показывать только в том случае, если за период от дня  $d_{min}$  до дня  $d_{max}$  включительно температура принимала или примет значение от  $t_{min}$  до  $t_{max}$  включительно. При этом гарантируется, что  $d_{min} \leq 0 \leq d_{max}$ , то есть диапазон дней должен включать сегодняшний.

Обратите внимание, что **температура не дискретна!** Например, если вчера значение температуры было  $-5$ , а сегодня  $-2$ , то необходимо считать, что в период, включающий сегодняшний и вчерашний дни, температура принимала не только значения  $-5$  и  $-2$ , но и  $-4$ , и  $-3$  (и даже  $-3,14$ , но в этой задаче все входные данные целочисленны).

Для каждого рекламного объявления определите, может ли оно быть показано в текущий день с учётом настроек и известных погодных данных.

## Формат ввода

В первой строке входного файла вводится единственное целое неотрицательное число  $N$  ( $N \leq 100\,000$ ).

В следующей строке вводится  $2N+1$  целых чисел — значения температуры в дни  $-N$ ,  $-N+1$ , ...,  $-1$ ,  $0$ ,  $1$ , ...,  $N$ . Значения температуры по модулю не превосходят  $10^9$ .

В третьей строке содержится единственное целое положительное число  $B$  ( $B \leq 100\,000$ ) — количество рекламных объявлений.

Наконец, в каждой из следующих  $B$  строк содержатся 4 целых числа  $t_{min}$ ,  $t_{max}$ ,  $d_{min}$ ,  $d_{max}$ , описывающие настройки соответствующего объявления. Гарантируется, что  $-10^9 \leq t_{min} \leq t_{max} \leq 10^9$  и  $-N \leq d_{min} \leq 0 \leq d_{max} \leq N$ .

## Формат вывода

Для каждого из объявлений выведите в отдельной строке «yes», если оно может быть показано согласно настройкам, и «no» в противном случае.

## Пример

Ввод	Вывод
2	yes
2 4 5 -2 2	yes
4	no
-1 -1 -1 1	no
0 10 -1 0	
6 8 -2 2	
3 3 -1 0	

Язык Mono C# 5.2.0

[Набрать здесь](#) [Отправить файл](#)

```
1 using System;
2
3 namespace B.Temperature
4 {
5     internal class Program
6     {
7         public static void Main(string[] args)
8         {
9             var n = int.Parse(Console.ReadLine());
10
11             var temperature = new int[2 * n + 1];
12             var tempStrings = Console.ReadLine().Split();
13             for (var i = 0; i < temperature.Length; i++)
14                 temperature[i] = int.Parse(tempStrings[i]);
15
16             var minMax = new MinMax[temperature.Length];
17             minMax[n] = new MinMax() {Max = temperature[n], Min = temperature[n]};
18             for (var i = n - 1; i >= 0; i--)
19                 minMax[i] = new MinMax(minMax[i + 1], temperature[i]);
20             for (var i = n + 1; i < minMax.Length; i++)
21                 minMax[i] = new MinMax(minMax[i - 1], temperature[i]);
22
23             var b = int.Parse(Console.ReadLine());
24             for (var i = 0; i < b; i++)
25             {
26                 var parts = Console.ReadLine().Split();
27
28                 var tmin = int.Parse(parts[0]);
29                 var tmax = int.Parse(parts[1]);
30                 var dmin = int.Parse(parts[2]) + n;
31                 var dmax = int.Parse(parts[3]) + n;
32
33                 var minTemp = Math.Min(minMax[dmin].Min, minMax[dmax].Min);
34                 var maxTemp = Math.Max(minMax[dmin].Max, minMax[dmax].Max);
35
36                 var yes = minTemp <= tmax && maxTemp >= tmin;
37                 Console.WriteLine(yes ? "yes" : "no");
38             }
39         }
40     }
41 }
```

[Отправить](#)

[Предыдущая](#)

[Следующая](#)