

NOIP模拟赛

By jiedai, nudun

一. 题目概况

中文题目名称	插队	比赛	采油区域
英文题目与子目录名	queue	match	oil
可执行文件名	queue	match	oil
输入文件名	queue.in	match.in	oil.in
输出文件名	queue.out	match.out	oil.out
每个测试点时限	1 秒	1 秒	2 秒
测试点数目	20	20	20
每个测试点分值	5	5	5
附加样例文件	有	有	有
结果比较方式	全文比较（过滤行末空格及文末回车）		
题目类型	传统	传统	传统
运行内存上限	128M	128M	128M

二. 提交源程序文件名

对于 C++语言	queue.cpp	match.cpp	oil.cpp
对于 C 语言	queue.c	match.c	oil.c
对于 pascal 语言	queue.pas	match.pas	oil.pas

插队（queue）

【问题描述】

一年一度的猴王争霸赛的报名开始啦！

目前有 n 只猴子排起了长长的队伍，每只猴子都有自己的身高 h 。当然，每只猴子都想着排到前面去，于是它们就要插队啦。

每次插队，插队的猴子都会去找目前排在它前面的那只猴子。如果前面那只猴子身高比自己矮，要插队的猴子就会直接用武力解决，从而排到它的前面，即所谓“猴高猴有理”。而对于那些身高比自己高的猴子，就不能直接插队了 T_T。

根据猴国的传统，只要抱在一只猴子的身上并用香蕉贿赂它，就能排到它前面了.....然而由于香蕉数量有限，每只猴子最多只能贿赂一只身高比自己高的猴子。

在开始插队之前，每只猴子都想知道，如果只有它一只猴在插队，它最多能排到哪个位置。

【输入格式】

输入文件名为 **queue.in**。

第一行输入一个正整数 n ，表示当前排队的猴子的数量。

第二行输入 n 个正整数 h ，表示排在第 i 位置的猴子的身高。

题目保证每只猴子的身高不超过 10^9 ，且每只猴子的身高各不相同。

【输出格式】

输出文件名为 **queue.out**。

输出 n 行，第 i 行输出一个整数表示只有第 i 只猴子在插队，它最多能排到哪个位置。

【输入输出样例 1】

queue. in	queue. out
7	1
7 5 2 6 3 4 1	1
	2
	1
	3
	3
	6

【样例 1说明】

第一只猴子已经排在队首了，第二只猴子贿赂了第一只猴子，也能排在队首。

第三只猴子贿赂了第二只猴子排在第二个位置。

第四只猴子身高比第三只猴子和第二只猴子高，可以直接插队，再贿赂第一只猴子，就可以排在队首了。

第五只猴子和第六只猴子贿赂了第四只猴子，都可以排在身高为5的第二只猴子后面。

第七只猴子贿赂了第六只猴子排在第六个位置。

【子任务】

对于 50%的测试点， $n \leq 5000$ 。
对于 90%的测试点， $n \leq 200000$ 。
对于 100%的测试点， $n \leq 1000000$ 。

比赛（match）

【问题描述】

刚刚成年的小猴终于可以参加猴王争霸赛了，于是小猴要前往花果山参加比赛。

猴国有 n 座山，其中 1 号山为花果山。有 $n-1$ 条山路连接着 n 座山，每座山都能通过一条唯一的路径到达花果山。

然而这是小猴第一次参加比赛，小猴并不知道通往花果山的道路，于是它只能够买地图来摸索路径。猴国一共有 m 种地图，对于第 i 种地图，它只能在第 x_i 座山才能买到，它可以指明从第 x_i 座山开始，通往花果山的 k_i 条山路，但是它需要花费 v_i 个香蕉。

小猴想知道从一些山出发，到达花果山最少所需要的香蕉数。现在它问了你 q 个问题，第 i 个问题是从第 w_i 座山出发，**不往回走地**到达花果山，至少需要多少个香蕉。

【输入格式】

输入文件名为 **match.in**。

第一行输入三个正整数 n, m, q 。

接下来 $n-1$ 行，每行两个整数 a_i, b_i ，表示第 a_i 座山与第 b_i 座山之间有山路连接。

接下来 m 行，每行三个整数 x_i, k_i, v_i ，意义如题目描述所示。**注意： k_i 可能会大于第 x_i 座山到达花果山的山路数。**

接下来 q 行，每行一个整数 w_i ，表示询问的出发点。

数据保证询问的山一定有出售地图，且通过地图一定有方案能够到达花果山。

【输出格式】

输入文件名为 **match.out**。

输出 q 行，第 i 行输出从第 w_i 座山出发，至少需要多少香蕉才能到达花果山。

【输入输出样例 1】

match.in	match.out
7 7 3	10
3 1	22
2 1	5
7 6	
6 3	
5 3	
4 3	
7 2 3	
7 1 1	

2 3 5	
3 6 2	
4 2 4	
5 3 10	
6 1 20	
5	
6	
7	

【样例 1说明】

对于第 5 座山，只能购买一种地图，花费 10 个香蕉可以指明三条山路，这足够到达花果山了，因此总花费是 10 个香蕉。

对于第 6 座山，只能购买一种地图，花费 20 个香蕉可以指明一条山路。之后，在第 3 座山花费 2 个香蕉可以买到直接通达花果山的地图。总花费是 22 个香蕉。

对于第 7 座山，最省的方式为先买 3 个香蕉指明两条山路的地图，再在第三座山花费 2 个香蕉买可以直接通达花果山的地图。总花费是 5 个香蕉。

【子任务】

对于 20%的测试点， $n, m, q \leq 5000$ ， v_i 不超过 10000。

另有 20%的测试点， $n, m, q \leq 50000$ ，第 i 座山和第 $i+1$ 座山之间有山路连接。（ $i < n$ ）

对于 60%的测试点， $n, m, q \leq 50000$ 。

另有 20%的测试点， $n, m, q \leq 200000$ ，所有的地图都能直接到达花果山。

对于 100%的测试点， $n, m, q \leq 200000$ ； $x_i, k_i \leq n$ ； $v_i \leq 10^9$ 。

采油区域（oil）

【问题描述】

Siruseri 政府决定将石油资源丰富的 Navalur 省的土地拍卖给私人承包商以建立油井。被拍卖的整块土地为一个矩形区域，被划分为 $M \times N$ 个小块。

Siruseri 地质调查局有关于 Navalur 土地石油储量的估测数据。这些数据表示为 $M \times N$ 个正整数，即对每一小块土地石油储量的估计值。

为了避免出现垄断，政府规定每一个承包商只能承包一个由 $K \times K$ 块相连的土地构成的正方形区域。

AoE 石油联合公司由三个承包商组成，他们想选择三块互不相交的 $K \times K$ 的区域使得总的收益最大。例如，假设石油储量的估计值如下：

```

1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 8 8 8 8 8 1 1 1
1 8 8 8 8 8 1 1 1
1 8 8 8 8 8 1 1 1
1 1 1 1 8 8 8 1 1
1 1 1 1 1 1 8 8 8
1 1 1 1 1 1 9 9 9
1 1 1 1 1 1 9 9 9

```

如果 $K = 2$, AoE 公司可以承包的区域的石油储量总和为 100, 如果 $K = 3$, AoE 公司可以承包的区域的石油储量总和为 208。
AoE 公司雇佣你来写一个程序, 帮助计算出他们可以承包的区域的石油储量之和的最大值。

【输入格式】

输入文件名为 **oil.in**。

输入第一行包含三个整数 M, N, K , 其中 M 和 N 是矩形区域的行数和列数, K 是每一个承包商承包的正方形的大小 (边长的块数)。接下来 M 行, 每行有 N 个正整数表示这一行每一小块土地的石油储量的估计值。

【输出格式】

输入文件名为 **oil.out**。

输出只包含一个正整数, 表示 AoE 公司可以承包的区域的石油储量之和的最大值。

【输入输出样例 1】

oil.in	oil.out
<pre> 9 9 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 8 8 8 8 8 1 1 1 1 8 8 8 8 8 1 1 1 1 8 8 8 8 8 1 1 1 1 1 1 8 8 8 1 1 1 1 1 1 1 8 8 8 1 1 1 1 1 9 9 9 1 1 1 1 1 9 9 9 </pre>	208

【数据规模与约定】

数据保证 $K \leq M$ 且 $K \leq N$ 并且至少有三个 $K \times K$ 的互不相交的正方形区域。

对于 15% 的测试点, $M, N \leq 12$ 。

对于 30% 的测试点, $M, N \leq 40$ 。

对于 50% 的测试点, $M, N \leq 100$ 。

对于 70% 的测试点, $M, N \leq 500$ 。

对于 100% 的测试点, $M, N \leq 1500$ 。

地的石油储量的估计值是非负整数且 ≤ 500 。