

12.26小测试

测试时间 2021年12月26日 08: 00 ~ 11: 30

题目	数位DP	最长上升子序列	序列移除2	g(n)	游戏
文件名	digit.cpp	lis.cpp	removal.cpp	gn.cpp	game.cpp
输入输出	digit.in/.out	lis.in/.out	removal.in/.out	gn.in/.out	game.in/.out
时间限制	1s	1s	1s	1s	2s
空间限制	256MB	256MB	256MB	256MB	256MB
编译命令	-O2 -std=c++14	-O2 -std=c++14	-O2 -std=c++14	-O2 -std=c++14	-O2 -std=c++14

T1: 数位DP(digit)

文件名 `digit.cpp/.in/.out` , 时间限制1秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

问题描述

作为本场比赛的签到题, 一道数位DP模板题再适合不过了。

设 $s(x)$ 为正整数 x 的各个数位上数字之和。

例如: $s(197) = 1 + 9 + 7 = 17$ 。

定义:

$$f(x) = \begin{cases} x & (x < 10) \\ f(s(x)) & (x \geq 10) \end{cases}$$

例如: $f(197) = f(17) = 8$

你需要回答 Q 个询问:

$$\sum_{x=l_i}^{r_i} f(x)$$

输入格式

输入的第一行包括一个正整数 Q 。

接下来的 Q 行每行有两个正整数 l_i, r_i 。

输出格式

按顺序输出 Q 个询问的答案, 每个询问单独输出一行。

样例1

样例输入

```
4
9 13
44 45
1998 2018
1122334455 6677889900
```

样例输出

```
19
17
102
2777777228
```

数据范围

对于 10% 的数据, $1 \leq l_i \leq r_i \leq 9$

对于 30% 的数据, $1 \leq l_i \leq r_i \leq 1000$

对于 50% 的数据, $1 \leq l_i \leq r_i \leq 1000000$

对于另外 20% 的数据, $r_i - l_i \leq 1000$

对于 100% 的数据, $1 \leq Q \leq 100$, $1 \leq l_i \leq r_i \leq 2^{60}$

T2: 最长上升子序列(lis)

文件名 `lis.cpp/.in/.out` , 时间限制1秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

问题描述

对数列 a_1, a_2, \dots, a_n , 它的上升子序列指的是形如 $a_{p_1}, a_{p_2}, \dots, a_{p_k}$, 且满足 $0 < p_1 < p_2 < \dots < p_k \leq n$ 和 $a_{p_1} < a_{p_2} < \dots < a_{p_k}$ 的序列。

求出最长上升子序列的长度, 并输出其中字典序 **最大** 的一个。

输入格式

第一行输入 1 个整数 n 。

第二行输入 n 个整数 a_1, a_2, \dots, a_n 。

输出格式

第一行输出最长上升子序列的长度 k 。

第二行输出 k 个正整数, 表示字典序最大的最长上升子序列。

样例1

样例输入

```
6
4 5 3 1 2 6
```

样例输出

```
3
4 5 6
```

样例解释

长度达到最大值 3 的 LIS 有 [4 5 6] 和 [1 2 6], 前者字典序更大。

样例2

见下发文件

数据范围

对于 20% 的数据, $1 \leq n \leq 10$

对于 50% 的数据, $1 \leq n \leq 100$

对于 70% 的数据, $1 \leq n \leq 1000$

对于 100% 的数据, $1 \leq n \leq 5000$, $1 \leq a_i \leq n$

共有 40% 的数据, 保证 LIS 的方案唯一。

T3: 序列移除2(removal)

文件名 `removal.cpp/.in/.out` , 时间限制1秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

问题描述

给定序列 s_1, s_2, \dots, s_n 。

求从中删除恰好 m 个元素后, 能组成的字典序最小的序列。

输入格式

本题有多组测试数据, 请处理到文件末尾。

对每组数据, 第一行输入 3 个整数 n, m, k 。

第二行输入 n 个整数 s_1, s_2, \dots, s_n 。

输出格式

对每组数据, 输出字典序最小的结果。

样例1

样例输入

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

样例输出

```
1 1
2 1
```

样例2

见下发文件

数据范围

对于 20% 的数据, $2 \leq n \leq 100$, 单个文件内所有 n 之和不超过 10^4 。

对于 50% 的数据, $2 \leq n \leq 1000$, 单个文件内所有 n 之和不超过 10^5 。

对于 100% 的数据, $2 \leq n \leq 10^5$, $1 \leq s_i \leq k \leq 10$, $1 \leq m \leq \min(n - 1, 10)$, 单个文件内所有 n 之和不超过 10^6 。

T4: g(n)(gn)

文件名 `gn.cpp/.in/.out` , 时间限制1秒, 空间限制256MB , 编译开启 `-O2` 优化, 支持 C++14。

问题描述

设正整数 n 的质因数分解式为 $n = \prod_{i=1}^k p_i^{r_i}$, 定义函数 $f(n) = \max_{i=1}^k r_i$, 求 $g(n) = \sum_{d|n} d \times f(\frac{n}{d})$ 。

输入格式

本题有多组测试数据。

第一行输入 1 个正整数 T , 表示数据组数。

对每组数据, 输入 1 行 1 个整数 n 。

输出格式

对每组数据, 输出 $g(n)$ 。

样例1

样例输入

```
3
24
108
720720
```

样例输出

```
52
243
3875760
```

样例解释

$$g(24) = f(24) + 2f(12) + 3f(8) + 4f(6) + 6f(4) + 8f(3) + 12f(2) + 24f(1) = 3 + 2 * 2 + 3 * 3 + 4 * 1 + 6 * 2 + 8 * 1 + 12 * 1 + 24 * 0 = 52$$

样例2

见下发文件

数据范围

对于 10% 的数据, $1 \leq T \leq 10$, $2 \leq n \leq 100$

对于 20% 的数据, $1 \leq T \leq 100$, $2 \leq n \leq 1000$

对于 50% 的数据, $1 \leq T \leq 10^4$, $2 \leq n \leq 10^5$

对于 80% 的数据, $1 \leq T \leq 10^4$, $2 \leq n \leq 10^7$

对于 100% 的数据, $1 \leq T \leq 10^5$, $2 \leq n \leq 10^7$

T5: 游戏(game)

文件名 `game.cpp/.in/.out` , 时间限制2秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

问题描述

你在玩一款游戏, 它在 $n \times m$ 的阵列上进行, 初始时有一些位置上有可供停留的平台, 用 `.` 符号表示, `#` 则表示该位置为空。

当处于某个平台上时, 可以选择上下左右 (分别用 `UDLR` 表示) 中的某个方向进行一次移动, 这次移动会朝着选定的方向一直前进, 直到遇到某个平台时停止移动并停留于此。移动时, 必须保证该方向上存在可供停留的平台, 否则就会因飞出阵列而失败。

一个平台只能停留一次, 当从某平台离开时, 该平台会消失, 消失后的位置等价于 `#`。

该游戏还有一个规则是, 你不能进行 180° 的大转弯, 例如: 若上一步你往“左”走了, 这一步就不能往“右”走, 只能走“上”、“左”、“下”中的一个方向。

现在指定起点 `S` 和终点 `T`, 这两个位置都存在平台, 且起点位置的平台在离开起点后也会消失。在起点处, 你可以选择任意一个方向作为初始方向。

完美通关该游戏, 需要从起点 `S` 出发, 途经所有的平台, 最后停留在终点 `T` 处。

你学业繁忙, 没有时间具体思考每一关的完美解法, 请写一个程序来协助通关。

输入格式

本题有多组数据, 第一行输入数据组数 T , 对于每组数据:

第一行输入 2 个正整数 n, m 。

接下来 n 行, 每行输入长度为 m 的字符串, 描述初始阵列。

输出格式

对每组数据, 若存在完美通关的移动方案, 输出 `YES`, 并在下一行输出任意一种方案 (由 `UDLR` 构成的字符串); 若不存在输出 `NO`。

样例1

样例输入

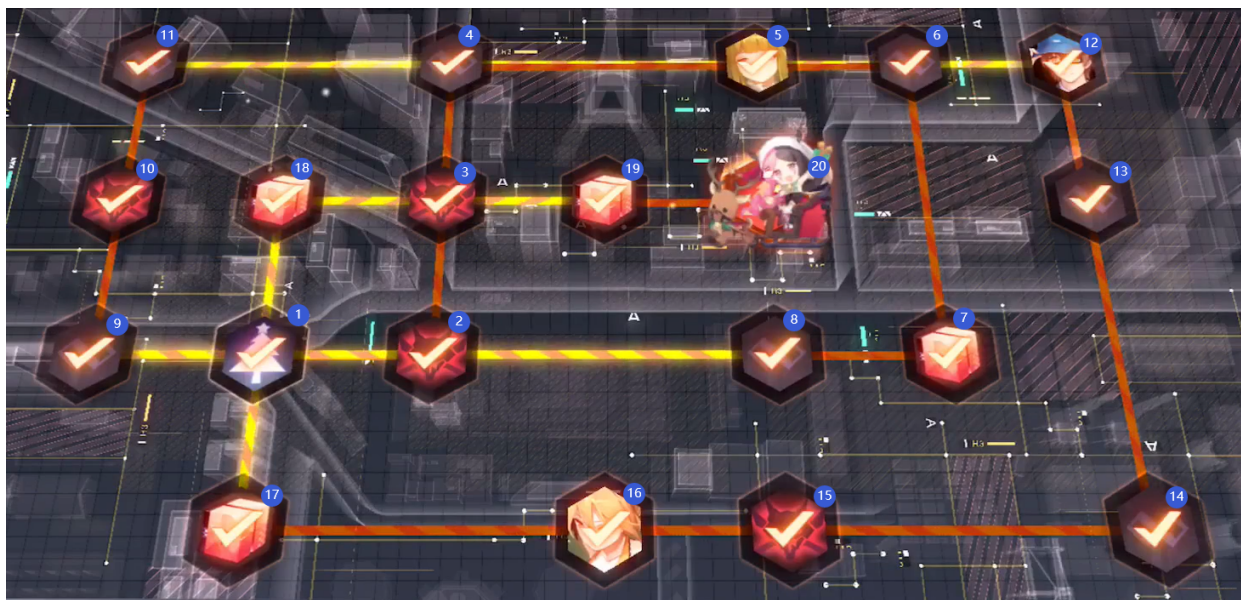
```
4
3 2
..
.S
#T
2 3
S..
#.T
1 3
.ST
4 7
.##...
....T#.
.S...#
#.#.#.
```

样例输出

```
YES
LURD
NO
NO
YES
RUURDLLUURDDLRLURR
```

样例解释

第四组数据的样例输出对应的方案如下图所示



数据范围

字符集 $\in \{ \#, s, t \}$, 保证有且只有 1 个起点 s 和终点 t 。

对于 10% 的数据, $2 \leq n \times m \leq 6$

对于 30% 的数据, $2 \leq n \times m \leq 20$

另有 30% 的数据, $n = 4, m = 7$

对于 100% 的数据, $1 \leq T \leq 10, 2 \leq n \times m \leq 30$