

递归搜索？

Void VS Int

状态 后效性

# 多重背包

- 多重背包问题：给定 $n$  种物品和一个背包。第 $i$ 种物品的价值是 $W_i$ ，其体积为 $V_i$ ，数量是 $K_i$ 件，背包的容量为 $C$ 。可以任意选择装入背包中的物品，求装入背包中物品的最大总价值。

二进制

转移依赖关系！

# 思考

每一个物品只有 $\text{least}[i]$ 到 $\text{most}[i]$ 个？

# 泛化，变种背包

从 $n$ 个可变长度的弹簧中取一些，第 $i$ 个弹簧的长度满足 $l[i] \sim r[i]$ ，获得的价值是基础价值 $a[i]$ 加额外 $(p - l[i]) * b[i]$ ， $p$ 为第 $i$ 个弹簧最终所取得的长度。

求价值最大！

$n, L \leq 1k, a, b \leq 1e9$

n个数，数值范围 $0 \sim 10^9$ ，从中取出任意多个数，求多少种方案使得和为C的倍数。

$$1 \leq N \leq 1e6$$

$$1 \leq C \leq 1000$$

## J - MU Puzzle [HDU - 4662](#)

Suppose there are the symbols M, I, and U which can be combined to produce strings of symbols called "words". We start with one word MI, and transform it to get a new word. In each step, we can use one of the following transformation rules:

1. Double any string after the M (that is, change Mx, to Mxx). For example: MIU to MIUIU.
2. Replace any III with a U. For example: MUIIIU to MUUU.
3. Remove any UU. For example: MUUU to MU.

Using these three rules is it possible to change MI into a given string in a finite number of steps?

### Input

First line, number of strings, n.

Following n lines, each line contains a nonempty string which consists only of letters 'M', 'I' and 'U'.

Total length of all strings  $\leq 10^6$ .

### Output

n lines, each line is 'Yes' or 'No'.

### Sample Input

```
2
MI
MU
```

### Sample Output

```
Yes
No
```



边界（敲黑板）

所有问题最终问到的问题

其中一眼可以看出答案的状态，  
二眼可以看出答案的状态，  
三眼.....  
看不出来，则不可做

滑雪

# 偏序

Sort (,,cmp)

## UVA 12260

$n(\leq 1000)$ 个物品，一个味道指数，一个是金钱指数，艾老师和数老师玩游戏，两人轮流取，每次一个。因为A老师非常憨厚，每次取味道最好的（相等时让对方收益最高），S老师很机智，取最终能让自己收益最高的（相等时让对方美味度最低）

$\text{sweet} \leq 1e9, \text{value} \leq 1e9$

最长子序列和

# Outline !

最优化                  //统计(概率)

状态(写搜索+记忆化)

转移(决策,excel大法好, 刷表法)

子问题 (结构)

边界!!! 敲黑板

# 最长公共子序列

HDOJ-1159:

Sample Input

abcfbc abfcab

programming contest

abcd mnp

Sample Output

4  
2  
0

## 辅助空间变化示意图

	a	b	c	f	b	c
a	1	1	1	1	1	1
b	1	2	2	2	2	2
f	1	2	2	3	3	3
c	1	2	3	3	3	4
a	1	2	3	3	3	4
b	1	2	3	3	4	4



## 子问题(结构)特征:

$$f(i,j)=\begin{cases} f(i-1,j-1)+1 & (a[i]==b[j]) \\ \max(f(i-1,j),f(i,j-1)) & (a[i]\neq b[j]) \end{cases}$$

- 由于 $f(i,j)$ 只和 $f(i-1,j-1)$ ,  $f(i-1,j)$ 和 $f(i,j-1)$ 有关, 而在计算 $f(i,j)$ 时, 只要选择一个合适的顺序, 就可以保证这三项都已经计算出来了, 这样就可以计算出 $f(i,j)$ . 这样一直推到 $f(\text{len}(a),\text{len}(b))$ 就得到所要求的解了.

- LCS

- $F[i][j]$ ?

# 最长上升子序列LIS

状态？

5种方法！

优化？

如果要求第 $k$ 项必须取呢？

# LCS

What ! ! ! !

第一个串长度为1000000

第二个串长度为1000