

## 第一題：這不是二分搜 (bsearch)

### 問題描述

喵喵最近剛學會了二分搜，但是他不想跟其他人一樣都搜正中間的位置，於是喵喵提出了以下的這個二分搜演算法：

令  $a_1, a_2, \dots, a_N$  是一個  $1, 2, \dots, N$  的排列，現在對於一組  $(L, R, x)$  ( $1 \leq L \leq R \leq N$ 、 $1 \leq x \leq N$ )，這個演算法的目標是想要找到「存不存在一個  $p \in [L, R]$  使得  $a_p = x$ 」。

1. 如果  $L > R$ ，那麼回傳  $-1$  並結束遞迴，代表這次搜尋失敗。
2. 隨機一個  $[L, R]$  區間內的正整數  $p$ 。
  - 如果  $a_p = x$ ，那麼回傳  $p$  並結束遞迴，代表這次搜尋成功。
  - 如果  $a_p > x$ ，那麼就將  $R$  變成  $p - 1$ ，代表接下來應該去搜尋左半邊。
  - 如果  $a_p < x$ ，那麼就將  $L$  變成  $p + 1$ ，代表接下來應該去搜尋右半邊。
3. 回到 (1.)。

本來，這個演算法在元素非嚴格遞增的陣列上是一定會正確的，不過笨笨的喵喵顯然忘記了這一點。他並沒有對  $a_L, a_{L+1}, \dots, a_R$  排序過就直接執行了他的二分搜演算法，導致有很多明明存在的數字卻沒有被找出來。這下，喵喵轉而開始好奇，給定一個排列，到底有多少個  $x$  可以在「無論上述演算法的步驟 (2.) 中隨機出來的  $p$  是多少」的情況下都能被正確的搜出來？

定義一個區間  $[L, R]$  的「二分搜度」 $B_{L,R}$  是代表有多少個  $x \in [1, N]$  可以永遠的被上述演算法回傳「搜尋成功」的結果。

因為喵喵懶得生一堆陣列，所以他想請你對於  $K = 1, 2, \dots, N$  分別輸出一個數字  $\text{ans}_K$ ，代表所有長度為  $K$  的區間之中二分搜度之和，也就是：

$$\text{ans}_K = \sum_{i=1}^{N-K+1} B_{i, i+K-1}$$

### 輸入格式

```
N  
a1 a2 ⋯ aN
```

- $N$  代表陣列的長度。
- $a_i$  代表陣列中第  $i$  項的數值 ( $1 \leq i \leq N$ )。

### 輸出格式

```
ans1 ans2 ⋯ ansN
```

- $ans_i$  代表所有長度為  $i$  的區間之中二分搜度之和 ( $1 \leq i \leq N$ )。

### 測資限制

- $1 \leq N \leq 1\,000\,000$ 。
- $a_1, a_2, \dots, a_N$  是  $1, 2, \dots, N$  的排列。
- 上面所有變數皆為整數。

### 範例測試

Sample Input	Sample Output
4 3 2 4 1	4 2 1 0

對於區間  $[L, R]$  可以被正確搜出來的  $x$  值分別有：

- $L = 1, R = 1, x = 3$ 。
- $L = 1, R = 3, x = 4$ 。
- $L = 2, R = 2, x = 2$ 。
- $L = 2, R = 3, x = 2$  或  $x = 4$ 。
- $L = 3, R = 3, x = 4$ 。
- $L = 4, R = 4, x = 1$ 。

其中，長度為 1、2、3、4 的分別有 4、2、1、0 個。

上述範例一符合子任務 2、3、4、5 跟 6 的限制。

Sample Input	Sample Output
10 1 2 3 4 5 6 7 8 9 10	10 18 24 28 30 30 28 24 18 10

上述範例二符合子任務 2、3、4、5 跟 6 的限制。

Sample Input	Sample Output
9 1 2 4 8 7 6 3 5 9	9 10 10 8 6 5 4 4 3

上述範例三符合子任務 2、3、4、5 跟 6 的限制。

### 評分說明

本題共有 6 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	1	$N \leq 3$ 。
2	10	$N \leq 15$ 。
3	14	$N \leq 60$ 。
4	17	$N \leq 600$ 。
5	22	$N \leq 5000$ 。
6	36	無額外限制。



## 第二題：這不是 LCS 問題 (lcs)

### 問題描述

『

給你兩個字串  $s$ 、 $t$ ，請找到他們的最長公共子序列 (longest common subsequence)。

』

這是動態規劃 (Dynamic Programming, DP) 的經典題。不過由於喵喵的耐心不好，不希望你輸出一大堆東西擠滿他的螢幕，喵喵也想要知道題目稍微變化一些你還能不能做出來，於是喵喵限制了你的輸出答案的長度。

具體來說，如果最長公共子序列的長度超過  $k$ ，你只要輸出任意一組長度為  $k$  的公共子序列就好。

### 輸入格式

```
 $n$   $m$   $k$   
 $s$   
 $t$ 
```

- $n$  是字串  $s$  的長度。
- $m$  是字串  $t$  的長度。
- $k$  代表目標的共同子字串長度。

### 輸出格式

```
 $l$   
 $C$ 
```

- $l$  是你找到的公共子序列的長度，必須滿足  $0 \leq l \leq k$ 。
- $C$  是你找到的公共子序列，其長度為  $l$ 。

### 測資限制

- $1 \leq n, m \leq 100\,000$ 。
- $|s| = n$ 。
- $|t| = m$ 。
- $s_i, t_j \in \{a, b, \dots, z\}$  ( $1 \leq i \leq |s|$ ,  $1 \leq j \leq |t|$ )。
- $1 \leq k \leq 100$ 。
- $k$  為整數。

### 範例測試

Sample Input	Sample Output
16 15 10 yagoothebestgirl butheisintheicu	6 thesti

$LCS(s, t) = \text{"thesti"}$ , 因為長度  $\leq k$  故直接輸出。

上述範例一符合子任務 1 跟 4 的限制。

Sample Input	Sample Output
10 10 1 aaaaabbbbb aaaaabbbbb	1 a

儘管  $LCS(s, t) = \text{"aaaaabbbbb"}$ , 但喵喵不需要長度超過  $k = 1$  的部分, 故輸出 "a" 或是 "b" 即可。

上述範例二符合全部子任務的限制。

Sample Input	Sample Output
1 1 1 x y	0

請注意可能會找不到非空的共同子序列。

上述範例三符合全部子任務的限制。

## 評分說明

對於一筆測試資料，你得到的**分數比重**  $S$  值如下：

- 若輸出的  $l$  不是最大值  $\min\{\text{LCS 長度}, k\}$ ，或是  $C$  不是長度為  $l$  且只包含小寫英文字母的字串，則  $S = 0.0$ 。
- 若輸出的  $l$  是最大值，而  $C$  不是  $s$  跟  $t$  的公共子序列，則  $S = 0.3$ 。
- 若輸出的  $l$  是最大值，且  $C$  是**任意一個**  $s$  跟  $t$  的公共子序列，則  $S = 1.0$ 。

本題共有 4 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，你的得分是該組所有測試資料之得分比重  $S$  中最低者，乘以該子任務的分數。

子任務	分數	額外輸入限制
1	38	$n, m \leq 1000$ 。
2	4	$k = 1$ 。
3	7	$k \leq 2$ 。
4	51	無額外限制。



### 第三題：這不是矩陣乘法 (matrix2)

#### 問題描述

喵喵在逛 Twitter 的時候滑到了許多有趣的數學巧合，像是  $\frac{1429}{6429} \cdot \frac{4287}{4286} = \frac{14\ 294\ 287}{64\ 294\ 286}$ 、一個三條邊長度分別為 45、97、56 的三角形面積是  $\sqrt{459\ 756}$ 、又或是  $\frac{3!7!6!6!4!9!8!5!}{4!6!8!7!3!8!6!4!8!} = \frac{376\ 664\ 985}{468\ 738\ 648}$  等等。

某一天，喵喵看到了如下關於矩陣乘法的巧合：

$$\begin{bmatrix} 7 & 8 \\ 4 & 3 \end{bmatrix} \times \begin{bmatrix} 9 & 4 \\ 2 & 7 \end{bmatrix} = \begin{bmatrix} 79 & 84 \\ 42 & 37 \end{bmatrix}$$

喵喵覺得這實在是太酷了！於是他想請你找出其他可能的解。

請找出所有符合如下算式且  $0 \leq a, b, \dots, g, h \leq 9$  的整數解  $(a, b, c, d, e, f, g, h)$ 。

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} 10a + e & 10b + f \\ 10c + g & 10d + h \end{bmatrix}$$

並將所有答案按照字典序排序過後輸出，每行為一組  $(a, b, c, d, e, f, g, h)$ ，輸出的 8 個整數之間沒有空格。

#### 輸入格式

本題沒有輸入。

#### 輸出格式

對每個解請輸出一行：

`abcdefgh`

- $(a, b, c, d, e, f, g, h)$  是上述算式的一組解。

### 測資限制

本題沒有輸入。

### 範例測試

Sample Input	Sample Output
<No Input>	00000000 37238728 59238926 78439427

上述範例輸出包含其中四組正確答案。請注意，只輸出上述四組並不會讓你得到任何分數。

### 評分說明

本題共有 1 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	100	無額外限制。

## 第四題：這不是簽到題 (registration3)

### 問題描述

在 ICPC 賽制下，答對題目的條件是必須通過所有的測試資料。只要答錯其中一筆，就是零分，沒有部份分數。在一些考量之下，便出現了一種設計目的是「所有隊伍或是幾乎所有隊伍都能答對」的題目，而這些題目被稱作是「簽到題」。

而現在，隨著高中資訊競賽越來越普及，在許多大大小小的比賽中，也都會有「簽到題」的存在。當你會用一種語言寫 Hello World，就能夠參加比賽來證明至少自己會寫一點程式，而辦比賽的單位也能證明你不是來騙參賽證明的。

下面就是一題簡單的簽不到題，請嘗試拿到這題的 AC 吧！

演講快要結束了，喵喵著急的環望四周，想要找出簽到表到底傳到哪裡了。

你聽到了喵喵可憐的請求，於是決定幫助他。你找出了演講廳的座位表，並且使用監視器找到了簽到表的所在。

現在給你一張  $30 \times 25$  的座位表，其中字元 M 代表喵喵，字元 P 代表簽到表。

你可以讓簽到表每秒鐘朝八方位移動一格，且喵喵在拿到簽到表之後要花費 2 秒鐘在上面簽名。請問喵喵最早什麼時候可以簽完簽到表？

可能有很多張座位表，而喵喵只要挑其中一張簽名就好了！

### 輸入格式

$s_{1,1}s_{1,2}s_{1,3} \cdots s_{1,25}$
$s_{2,1}s_{2,2}s_{2,3} \cdots s_{2,25}$
$\vdots$
$s_{30,1}s_{30,2}s_{30,3} \cdots s_{30,25}$

- $s_{i,j}$  代表第  $i$  排第  $j$  列的座位上的人 ( $1 \leq i \leq 30$ 、 $1 \leq j \leq 25$ )。

### 輸出格式

ans

- ans 代表喵喵最早還要幾秒才可以簽完簽到表。

### 測資限制

- $s_{i,j} \in \{a, b, \dots, z, A, B, \dots, Z\}$  ( $1 \leq i \leq 30$ 、 $1 \leq j \leq 25$ )。
- 保證一定只有一個 M 且至少有一個 P。

## 範例測試

Sample Input	Sample Output
OUbkGnSlefZPyYFrpsazvTBrC JYTvRpIbKsyomoEhtYzuCmTmY nrVDlnEVeRGxUUgIIZbsNpLrx ijERBQoGOnnakJVmVYZGZVsJB ARutYBIGfIVeRHdJWtUobRfU KaNeSjXVGLAyOuvlwJvXDpFHs BpgzXNSOXoHYbuPXeSvVFxHxL TwcSzbpNDWfdnwWrmUhCNGBRy kwfksUUNeVEraFSUKzVUOxfrh YkytNlipDGOnnAhjgmCxegcLE PxQgCaWngGNleTvtmWSJvYmHV IYFzRwExesmRiYoUZvNDxwJhk jymxqiCmptgmeUyDOWnEqHmiZ msTZImj jnqDxwzIfqmpNDPxBB imRoZiyhsQskDZHtuuTtTXLVK ySpgGcwrG0aoeYfXTEqetPQFb HpAzTWyEphnpmsnHRZEitYFC qMgWHTHXcBFpYYLXrsXRVKRU EXWnmNxmtJEFHpnJglwDBLYNq nguUBDrrOEPNsvgyCcIjqpsSo uWFzjnlTLcCshejLvzQNJBfKf UpVnwQeaBipKfyTpaAYSZppCm EmwSbLmfdRDDTgRpTummNHBWg LipylSlAfFEyISmYWIfeeQnQW GnvVnmSftFusuwxJliinvvrDk OEzSvpFHlUfYApSqXlenkCkhf BHppUzIkctfbAnhnQwprefppl kokWzsaHOLaFfeYisSocUTEZm XqmemZGSzUUoqgfJaNdxfhqIa eUNQyURixlGHtdhbLGdJklgoG	9

上述範例符合子任務 2 的限制。

## 評分說明

本題共有 2 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	50	只有一張簽到表。
2	50	無額外限制。

## 第五題：這不是序列操作 (sequence)

### 問題描述

給一個長度為  $n$  的序列  $a_1, a_2, \dots, a_n$ ，保證元素兩兩相異。喵喵想要對他做  $q$  次操作：

- 1：輸出  $\min\{a_1, a_2, \dots, a_n\}$ 、把他刪掉、並使  $n \leftarrow n - 1$ ，保證此時陣列  $a$  非空。
- 2：輸出  $\max\{a_1, a_2, \dots, a_n\}$ 、把他刪掉、並使  $n \leftarrow n - 1$ ，保證此時陣列  $a$  非空。
- 3：輸出  $\text{mex}\{a_1, a_2, \dots, a_n\}$ 、把他加到陣列  $a$  的最後端、並使  $n \leftarrow n + 1$ 。

其中， $\min\{c_1, c_2, \dots, c_k\}$  跟  $\max\{c_1, c_2, \dots, c_k\}$  的回傳值是  $c_1, c_2, \dots, c_k$  的最小值跟最大值，而  $\text{mex}\{c_1, c_2, \dots, c_k\}$  回傳的是最小沒有在  $c_1, c_2, \dots, c_k$  裡出現過的非負整數。舉例來說， $\text{mex}\{3, 2, 6\} = 0$ 、 $\text{mex}\{3, 0, 1, 4\} = 2$ 、而  $\text{mex}\{0\} = 1$ 。

為了減少輸出量，你只需要輸出  $\lceil Q/10 \rceil$  個數字，其中第  $k$  個數字  $\text{ans}_k$  代表第  $(10k - 9)$  次操作到第  $\min(10k, Q)$  次操作的答案之和。

### 輸入格式

```
N Q
a1 a2 ... aN
S
```

- $N$  代表陣列的長度。
- $Q$  代表操作的次數。
- $a_i$  代表陣列中第  $i$  項的數值，保證  $a_i$  兩兩相異 ( $1 \leq i \leq N$ )。
- $S$  是一個長度為  $Q$  的字串，第  $i$  個字元  $S_i$  代表第  $i$  次操作的指令 ( $1 \leq i \leq Q$ )。

### 輸出格式

```
ans1 ans2 ... ans $\lceil Q/10 \rceil$ 
```

- $\text{ans}_i$  是第  $(10i - 9)$  到  $\min(10i, Q)$  次操作的答案之和 ( $1 \leq i \leq \lceil Q/10 \rceil$ )。

### 測資限制

- $0 \leq N \leq 200\,000$ 。
- $1 \leq Q \leq 5\,000\,000$ 。
- $0 \leq a_i \leq 200\,000\,000$  ( $1 \leq i \leq N$ )。
- $a_i \neq a_j$  ( $1 \leq i < j \leq N$ )。
- $|S| = Q$ 。
- $S_i \in \{1, 2, 3\}$  ( $1 \leq i \leq Q$ )。
- 上面除  $S$  外之所有變數皆為整數。

### 範例測試

Sample Input	Sample Output
5 7 4 8 7 6 3 1323132	19

答案依序為 3、0、8、1、0、0、7。

上述範例一符合子任務 1 跟 5 的限制。

Sample Input	Sample Output
5 5 4 2 0 6 9 12212	21

答案依序為 0、9、6、2、4。

上述範例二符合子任務 1、2 跟 5 的限制。

Sample Input	Sample Output
6 8 31 4 15 9 26 200000000 32233323	200000068

答案依序為 0、200 000 000、31、1、2、3、26、5。

上述範例三符合子任務 1、3 跟 5 的限制。

Sample Input	Sample Output
0 4	0
3231	

上述範例四符合子任務 1、4 跟 5 的限制。

### 評分說明

本題共有 5 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	16	$N \leq 1000$ 、 $Q \leq 1000$ 。
2	17	$S_i \in \{1, 2\}$ ( $1 \leq i \leq Q$ )。
3	12	$S_i \in \{2, 3\}$ ( $1 \leq i \leq Q$ )。
4	29	$N = 0$ 。
5	26	無額外限制。



## 第六題：這不是排序 (sort2)

### 問題描述

這是世界上最單純的排序法：

```
void SimpleSort(std::vector<int> &A) {  
    int N = A.size();  
    for (int i = 0; i < N; ++i)  
        for (int j = 0; j < N; ++j)  
            if (A[i] < A[j]) std::swap(A[i], A[j]);  
}
```

有一天，可愛的拉菲在寫 code 的時候不小心把小於符號的開口方向搞錯了，於是就變成了這個樣子：

```
void NotSimpleSort(std::vector<int> &A) {  
    int N = A.size();  
    for (int i = 0; i < N; ++i)  
        for (int j = 0; j < N; ++j)  
            if (A[i] ^ A[j]) std::swap(A[i], A[j]);  
}
```

請問你，以這種方法排序完的結果是什麼？

### 輸入格式

$N$
$A_1 A_2 \cdots A_N$

- $N$  代表陣列的長度。
- $A_i$  代表陣列中第  $i$  項的數值 ( $1 \leq i \leq N$ )。

### 輸出格式

$A'_1 A'_2 \cdots A'_N$
-------------------------

- $A'_i$  代表經過 `NotSimpleSort()` 排序後陣列  $A$  的第  $i$  項 ( $1 \leq i \leq N$ )。

### 測資限制

- $1 \leq N \leq 100\,000$ 。
- $0 \leq A_i \leq 2^{31} - 1$  ( $1 \leq i \leq N$ )。
- 上面所有變數皆為整數。

### 範例測試

Sample Input	Sample Output
3	1 2 3
2 3 1	

上述範例一符合全部子任務的限制。

Sample Input	Sample Output
2	1 2
1 2	

上述範例二符合全部子任務的限制。

### 評分說明

本題共有 2 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	30	$N \leq 500$ 。
2	70	無額外限制。