

110 學年度全國資訊學科能力競賽

臺南一中 校內初選 試題本

競賽說明

1. 競賽時間：2021/09/10 13:30 ~ 16:30，共 3 小時。
2. 本次競賽試題共 6 題，每題皆有子任務。
3. 每題的分數為該題所有子任務得分數加總；單筆子任務得分數為各筆繳交在該筆得到的最大分數。
4. 競賽系統：<http://210.70.137.160/contest/>。
5. 競賽記分板：<http://210.70.137.160/ranking/>。
6. 本次初選比照南區賽提供記分板，複選比照全國賽不提供記分板。
7. 全部題目的輸入皆為標準輸入。
8. 全部題目的輸出皆為標準輸出。
9. 所有輸入輸出請嚴格遵守題目要求，多或少的換行及空格皆有可能造成裁判系統判斷為答案錯誤。
10. 每題每次上傳間隔為 120 秒，裁判得視情況調整。
11. 所有試題相關問題請於競賽系統中提問，題目相關公告也會公告於競賽系統，請密切注意。
12. 如有電腦問題，請舉手向監考人員反映。
13. 如有廁所需求，須經過監考人員同意方可離場。
14. 不得攜帶任何參考資料，但競賽系統上的參考資料可自行閱讀。
15. 不得自行攜帶隨身碟，如需備份資料，請將資料儲存於電腦 D 槽。
16. 競賽中請勿交談。請勿做出任何會干擾競賽的行為。

A. ABC 體操

Problem ID: Gymnastics

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: 裏道大哥哥和排成一列的小朋友，出自動畫《陰晴不定的體操哥哥》

「裏道大哥哥」是兒童律動節目《和媽媽在一起》的主持人，今天又到了 ABC 體操的時間。小朋友們已經迫不及待地「**按照身高從矮到高排成一列**」，只不過裏道大哥哥今天想要來點特別的。

裏道大哥哥想要將小朋友們分組，只不過由於工作上太累的關係，便想偷懶地直接將排成一列的隊伍切成幾段，因此每個組別都是隊伍中的連續區間。

裏道大哥哥這時又想知道使用這樣的分組方法的話，各組的小朋友身高的中位數分別是多少。

但又是因為工作上太累的關係，裏道大哥哥決定將計算的工作交給你。

為了方便你們溝通，裏道大哥哥會從隊伍的「某一邊」開始為小朋友編號 $1, 2, \dots, N$ 直到另外一邊。

接著多次會詢問你編號 X 至編號 Y 的小朋友分成一組的話（包含 X 及 Y ），該組小朋友身高的「中位數」。

由於不敢違逆前輩，身為裏道大哥哥後輩的你只好乖乖幫忙計算了。

— 輸入 —

第一行有 1 個正整數 N ，代表小朋友的人數。

第二行有 N 個整數 A_1, A_2, \dots, A_N ，編號 i 的小朋友的身高為 A_i 。

第三行有 1 個正整數 Q ，表示接下來有 Q 筆詢問。

接下來 Q 行，每行有 2 個正整數 X, Y ，表示詢問編號 X 至編號 Y 的小朋友分成一組時的身高中位數是多少。

— 輸出 —

對於每筆詢問，輸出該區間的身高中位數。

— 輸入限制 —

- $N \leq 10^5$
- $1 \leq A_i \leq 10^9$
- $Q \leq 10^5$
- $Y - X$ 為 2 的倍數

— 子任務 —

| 編號 | 分數 | 額外限制 |
|----|----|---|
| 0 | 0 | 範例輸入輸出 |
| 1 | 10 | $A_i = i$ 且 $X \leq Y$ |
| 2 | 30 | $A_i \leq A_{i+1}$ 且 $X \leq Y$ 且 $N \leq 10$ |
| 3 | 10 | $A_i \leq A_{i+1}$ 且 $X \leq Y$ |
| 4 | 10 | $A_i \leq A_{i+1}$ |
| 5 | 40 | 無額外限制 |

— 範例輸入 —

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

— 範例輸出 —

```
3
4
```

B. 國中會考分發

Problem ID: Komachi

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: 小町錄取總武高中，出自動畫《果然我的青春戀愛喜劇搞錯了。完》

又來到了國中會考的季節，而今天是放榜的日子，小町如願以償地進到總武高中。

不過說到放榜，首先就要了解學校分發的方式。

每個學生經過考試後都會得到一個總成績，並且可以填寫五個志願。而分發的方式是依照成績高低決定優先順序，接下來從最高的志願序開始依序檢查學校的名額是否額滿，如果還沒額滿，就表示順利考進該所學校。

假如每間學校都只能收 k 位學生，告訴你 n 個學生的成績由高到低排序的結果，以及每個學生填寫的志願序，你能知道這 m 間學校的最終錄取人數以及錄取的學生編號分別為何嗎？

— 輸入 —

第一行有三個整數 n, m, k ，代表有 n 位學生， m 間學校，每間學校收 k 個人。

第二行是一個長度為 n 的排列，代表學生分數由高到低的排名結果。

接下來有 n 行，每行有五個數字，其中第 i 行代表編號為 i 的學生選填的五個志願。

— 輸出 —

輸出 m 行，每行的第一個數字 a_i 代表進入第 i 間學校的人數，接下來輸出 a_i 個數字，代表進入第 i 間學校的學生編號，編號請由小到大排序。

— 輸入限制 —

- $1 \leq n \leq 200000$
- $5 \leq m \leq 1000$
- $1 \leq k \leq n$

— 子任務 —

| 編號 | 分數 | 額外限制 |
|----|----|-----------------------|
| 0 | 0 | 範例輸入輸出 |
| 1 | 8 | $k = n$ |
| 2 | 12 | $m = 5$, 每個人填的志願順序相同 |
| 3 | 56 | $n \leq 5000$ |
| 4 | 24 | 無額外限制 |

— 範例輸入 1 —

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

— 範例輸出 1 —

```
2 2 3
1 1
0
0
0
```

— 範例輸入 2 —

```
8 7 2
7 5 4 6 3 2 8 1
4 5 7 6 3
2 3 4 1 5
2 3 4 1 5
1 7 6 5 4
1 2 3 4 5
1 3 2 4 5
1 6 4 3 5
2 3 1 4 5
```

— 範例輸出 2 —

```
2 5 7
2 2 3
2 6 8
1 1
0
0
1 4
```

C. GAMAGAMA

Problem ID: GAMAGAMA

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: 風花向遊客解說海洋生物，出自動畫《白沙的 Aquatope》

GAMAGAMA 水族館面臨即將停業的危機，代理館長空空琉決定推出觸摸池的活動來吸引遊客，由前偶像團體成員的風花擔任導覽員。

觸摸池吸引了不少家長帶著小孩來參觀，風花推出了有獎徵答，答對的小朋友累積積分來兌換獎品。

而空空琉為了炒熱氣氛，決定加碼讓遊客的積分能夠翻倍。

GAMAGAMA 水族館有兩條觀察大型水族箱的海底隧道，空空琉會在海底隧道中沿路擺上調整積分的告示牌，遊客要選擇其中一條隧道參觀，每走過一面告示牌，就會按照告示牌上的內容對目前的積分進行修改。

例如你目前的積分是 x ，則經過 $+a$ 的告示牌時，積分就會變成 $x + a$ ；經過 $\times a$ 的告示牌時，積分就會變成 $x \times a$ 。積分的修改是經過告示牌後立即運算更新，不是四則運算。

為了控制不要讓積分過度膨脹，必須先進行模擬。

一開始兩條隧道都是沒有擺放任何告示牌，隨著活動進行，會不斷在其中一條隧道的尾端加上一面告示牌。

要模擬的變動及詢問請見輸入說明。

— 輸入 —

第一行有一個數字 Q ，表示接下來的操作次數。

接下來有 Q 行，每一行為操作為以下其一（皆不含引號）。

- “1 k ” — 若遊客進入隧道前的積分為 k ，請輸出經過兩條隧道後的積分較大值。
- “2 $i + a$ ” — 在 i 號隧道最尾端加上 $+a$ 的告示牌。
- “2 $i * a$ ” — 在 i 號隧道最尾端加上 $\times a$ 的告示牌。
- “3” — 找出是否存在遊客進入隧道前的積分為整數 m ，在經過兩條隧道後的積分相同。若非整數解、不存在或多組解則輸出 “no”（不含引號）。

— 輸出 —

對於每筆操作 1，請輸出一個整數 k 。

對於每筆操作 3，請輸出一個整數 m 或 “no”（不含引號）。

— 輸入限制 —

- $1 \leq Q \leq 10^6$
- 操作 1 中 $-10000 \leq k \leq 10000$
- 操作 2 中 $i \in \{1, 2\}$ 且 $-100 \leq a \leq 100$
- 保證若初始積分在 $[-10000, 10000]$ 之內，則無論選擇哪條隧道，任何時間手上的積分都在 $[-10^9, 10^9]$ 之內。

— 子任務 —

| 編號 | 分數 | 額外限制 |
|----|----|----------------------------|
| 0 | 0 | 範例輸入輸出 |
| 1 | 18 | $Q \leq 8000$, 且不含操作 3 |
| 2 | 8 | 操作 2 只有 $+$ |
| 3 | 14 | 操作 1 的初始積分為 0 或 1, 且不含操作 3 |
| 4 | 39 | 不含操作 3 |
| 5 | 21 | 沒有其他限制 |

— 範例輸入 1 —

8
 1 3
 2 1 + 2
 1 3
 2 2 * 2
 1 3
 2 1 + 2
 1 3
 3

— 範例輸出 1 —

3
 5
 6
 7
 4

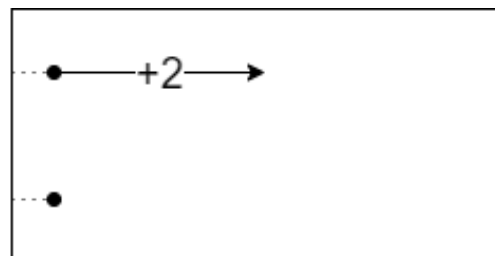
— 範例說明 1 —



(一開始兩條隧道沒有告示牌)

操作 1：3 $[\max(3, 3) = 3]$

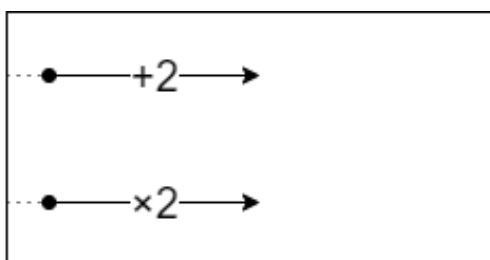
\Rightarrow



操作 2：隧道 1 接上 +2

操作 3：5 $[\max(3 + 2, 3) = 5]$

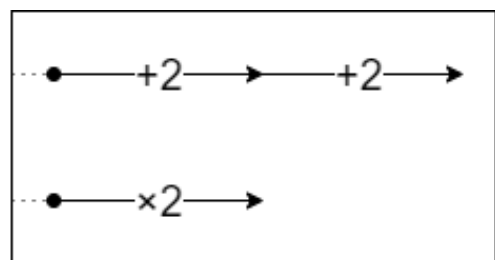
\Rightarrow



操作 4：隧道 2 接上 $\times 2$

操作 5：6 $[\max(3 + 2, 3 \times 2) = 6]$

\Rightarrow



操作 6：隧道 1 接上 +2

操作 7：7 $[\max(3 + 2 + 2, 3 \times 2) = 7]$

操作 8：4 $[4 + 2 + 2 = 4 \times 2 = 8]$

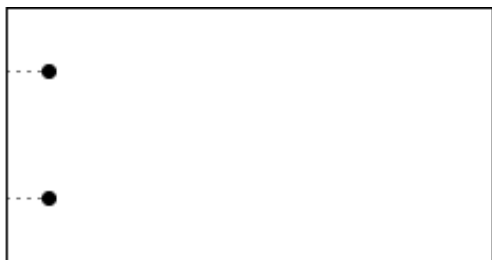
－ 範例輸入 2 －

```
9
1 3
2 1 * 0
1 3
3
2 2 * 0
1 3
3
2 1 + 2
3
```

－ 範例輸出 2 －

```
3
3
0
0
no
no
```

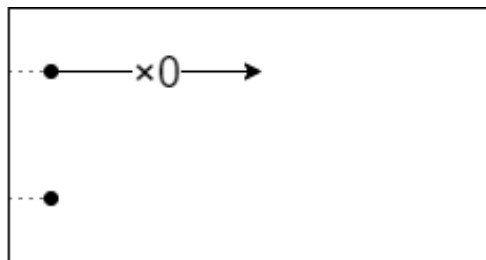
－ 範例說明 2 －



(一開始兩條隧道沒有告示牌)

操作 1: 3 $[\max(3, 3) = 3]$

⇒

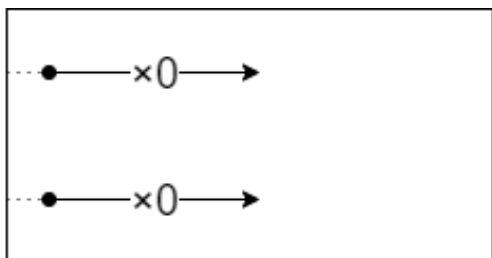


操作 2: 隧道 1 接上 $\times 0$

操作 3: 3 $[\max(3 \times 0, 3) = 3]$

操作 4: 0 $[0 \times 0 = 0 = 0]$

⇒

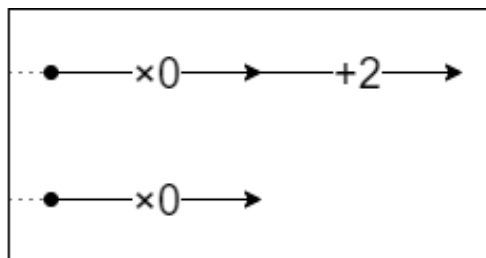


操作 5: 隧道 2 接上 $\times 0$

操作 6: 0 $[\max(3 \times 0, 3 \times 0) = 0]$

操作 7: no [多組解]

⇒



操作 8: 隧道 1 接上 $+2$

操作 9: no [不存在]

D. 花子一口吞

Problem ID: EatAllLeftMost

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: 在你寫 code 的期間，花子越來越餓，你可能得快點 AC...，出自動畫《萌獸寵物店》

自稱為半龍人的貪吃少女——花子，實際上卻是魔界四大公爵-龍族法夫納家的下任家主！因為厭倦了家中殘酷的訓練還有一成不變的食物，於是離家出走來到了人類源藏開的魔獸寵物店當起了店員。

作為寵物店店員，最重要的工作就是給每個寵物分配飼料，每到開飯時間，花子便要為每個飼料盆分配不同的飼料量。今天中午，花子又一次完美的做好了分配工作，突然之間她覺得肚子有些餓了，可能是因為早上只吃了十碗飯，實在不太夠，於是她決定偷吃一點寵物們的飼料，作為自己辛苦工作的獎勵。

「但如果又偷吃飼料被店長發現的話，肯定又會被揍了...該怎麼辦呢？」花子沈思了許久，想起了好久沒做的「一口吞」！

「一口吞」是花子獨創的吃法：她會選定某盆飼料開始一顆顆往嘴巴裡吃，一但到達嘴巴能裝的最多數量 K ，就會一口吞下！如果盆子裡還有剩，便會繼續將飼料裝入嘴巴裡，一但到達嘴巴能裝的最多數量 K ，就會一口吞下...如此反覆循環，直到整盆飼料都空了後，就往下一個飼料盆繼續吃，還沒吞下去的飼料當然也還裝在嘴巴裡！然後繼續吃...吃...吃...吃到滿意為止。

「如果這次又每盆都偷吃一點，就算只少了那麼一點點，肯定還是會被店長發現而被挨罵...對了！只要把連續幾個飼料盆都吃光，假裝本來就沒有裝飼料就好了！而且還想要藏一點飼料在嘴巴裡當作點心...」

身為魔界菁英的她想出了如此辦法，她想要知道自己得從哪一盆開始吃，連續吃到哪一盆為止，可以使她嘴巴裡剩的飼料量最大，就能完美解決這個難題，而她會將剩餘的飼料從嘴巴裡吐出來作為點心，等一下才不會餓肚子呢！

另外如果吃完最後一盆的時候恰好達到嘴巴容量 K ，她仍然會忍不住吞下去，而無法作為點心。

你不需要在意這個策略是否會被店長發現，因為這位龍族少女要求你幫她找到答案，不然你可能會成為那些飼料的替代品！為了你自己的安危著想，請幫助花子找到留下最多飼料當作點心的方法。

— 輸入 —

第一行有 2 個正整數 N 、 K ，代表寵物飼料的數量和花子嘴巴的容量。

第二行有 N 個正整數 a_1, a_2, \dots, a_N ，代表每份寵物飼料的量。

— 輸出 —

輸出花子最多能留下多少飼料當作點心。

— 輸入限制 —

- $1 \leq N \leq 5 \cdot 10^5$
- $1 \leq K \leq 10^9$
- $1 \leq a_i \leq 10^9$

— 子任務 —

| 編號 | 分數 | 額外限制 |
|----|----|--|
| 0 | 0 | 範例輸入輸出 |
| 1 | 4 | $N \leq 300$ |
| 2 | 9 | $N \leq 8000$ |
| 3 | 31 | $K \leq 30$ |
| 4 | 23 | 符合 $a_i \neq a_{i-1}$ 的 i 最多 2 個，且 $K \leq 2000$ |
| 5 | 33 | 無額外限制 |

— 範例輸入 1 —

5 5
1 1 1 1 1

— 範例輸出 1 —

4

— 範例說明 1 —

其中一個最大的區間為 $[1, 4]$ ，留下的量為 $1 + 1 + 1 + 1 = 4$ 。
注意若花子吃了 $[1, 5]$ ，則她會將其整口吞掉而留下 0。

— 範例輸入 2 —

7 11
3 3 5 7 7 8 8

— 範例輸出 2 —

8

— 範例說明 2 —

其中一個最大的區間為 $[6, 6]$ ，留下的量為 8。

— 範例輸入 3 —

5 5
1 2 3 2 1

— 範例輸出 3 —

4

— 範例說明 3 —

最大的區間為 $[1, 5]$ 。
花子先吃了 a_1, a_2 ，在吃到 a_3 第 2 顆後一口吞下，再吃了剩下的 1 顆（留下 1）。
最後吃了 a_4, a_5 ，留下的量為 $1 + 2 + 1 = 4$ 。

E. 蓋歐格

Problem ID: Georg

Time Limit: 5.0s

Memory Limit: 512MiB



Figure 1: 利姆路等人正在討論使用傳送魔法來傳送軍隊的方法，出自動畫《關於我轉生成史萊姆這檔事第二季》

為了阻止克雷曼的陰謀，朱拉·坦派斯特聯邦國決定派出大軍，然而要將龐大的軍隊快速地送到目的地可不是一件簡單的事。

利姆路決定兵分多路，在連接不同城鎮上的道路上分別施展魔法建立單向「傳送通道」來形成「傳送網路」。

如此強大的傳送魔法勢必要有一些條件才能施展：

- 傳送通道為單向的。
- 傳送通道連接兩個城鎮，可以從城鎮 A 傳送到城鎮 B 。
- 兩個城鎮間可以施展多條傳送通道。
- 為了防止軍隊迷路，從某個城鎮離開之後，必定無法回到該城鎮。
- 任意選擇傳送通道前進，最終必定都能到達目的地。

為了讓傳送網路的效益最大化，利姆路想出了一套方法來評估傳送網路的效率。

每條傳送通道都有一個數值「蓋歐格」來表示該通道的效率，而整個傳送網路則必定可以透過以下方式簡化來計算整體的「蓋歐格」。

一個傳送網路只有一個起點及一個終點，且必定符合以下情況之一（參考 Figure 2）：

- 情形 1：傳送網路恰好就是一條傳送通道。
- 情形 2：從城鎮 A 到城鎮 B 的傳送網路可以拆分為兩個子傳送網路 X 及 Y ， X 的起點為 A ，終點為中間城鎮 C ； Y 的起點為中間城鎮 C ，終點為城鎮 B 。網路 X 的蓋歐格為 x ，網路 Y 的蓋歐格為 y ，則城鎮 A 到城鎮 B 整體的蓋歐格為 $x + y$ 。
- 情形 3：從城鎮 A 到城鎮 B 的傳送網路可以拆分為兩個子傳送網路 X 及 Y ， X 和 Y 的起點皆為 A ，終點皆為 B 。網路 X 的蓋歐格為 x ，網路 Y 的蓋歐格為 y ，則城鎮 A 到城鎮 B 整體的蓋歐格為 $\frac{1}{\frac{1}{x} + \frac{1}{y}}$ 。

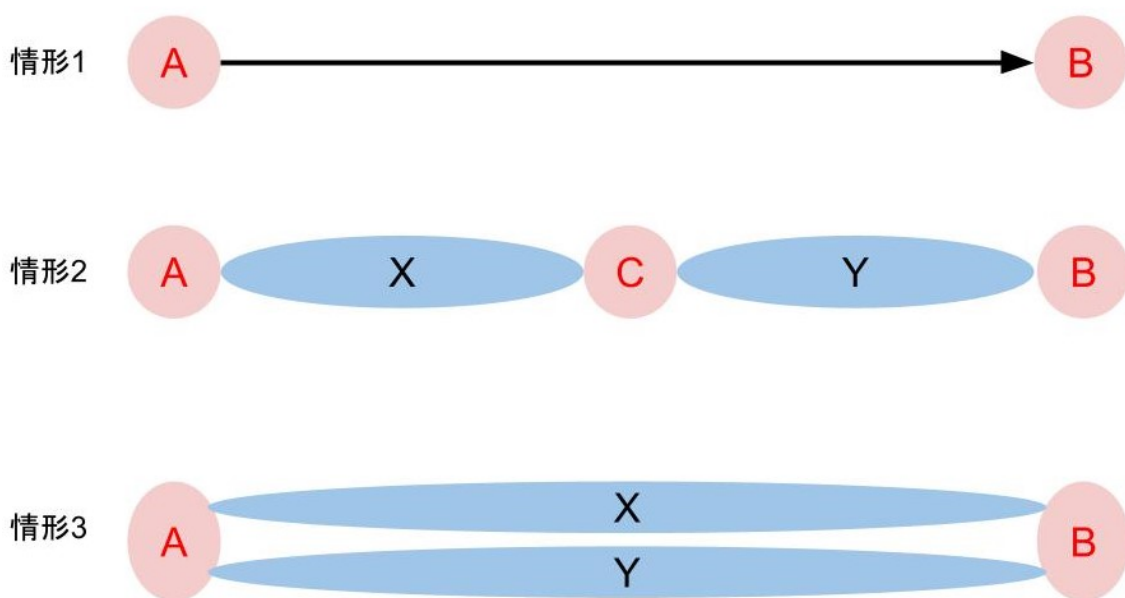


Figure 2: 傳送網路的組成情形， A 及 B 為城鎮，黑色箭頭為傳送通道， X 及 Y 為傳送網路，其中可能包含多個傳送通道及多個城鎮

— 輸入 —

輸入的第一行包含兩個正整數 N 、 E ， N 為城鎮數量， E 為傳送通道數量。

接下來有 E 行，每行包含三個正整數 A 、 B 、 R ，表示傳送通道可以從城鎮 A 傳送到城鎮 B ，其「蓋歐格」為 R 。

— 輸出 —

請輸出一行，以 1 號城鎮為起點，2 號城鎮為目的地的傳送網路，整體「蓋歐格」為多少？

— 輸入限制 —

- $1 \leq N, E \leq 10^6$
- $1 \leq A, B \leq N$
- $1 \leq R \leq 30$

— 評測說明 —

本題採用 special judge，只要輸出與答案的絕對誤差或相對誤差 $\leq 10^{-6}$ 就判定為通過。也就是說假設你的輸出是 O ，答案是 A ，若 $\min(|O - A|, \frac{|O - A|}{A}) \leq 10^{-6}$ 就會通過。

— 子任務 —

| 編號 | 分數 | 額外限制 |
|----|----|----------------------|
| 0 | 0 | 範例輸入輸出 |
| 1 | 6 | 傳送網路的組成只有情形 1 及 2 |
| 2 | 7 | 傳送網路的組成只有情形 1 及 3 |
| 3 | 51 | $0 < N, E \leq 1000$ |
| 4 | 36 | 無額外限制 |

— 範例輸入 —

```

3 5
1 2 17
1 2 15
3 2 30
1 3 20
1 3 10

```

— 範例輸出 —

```

6.546091015

```

— 範例說明 —

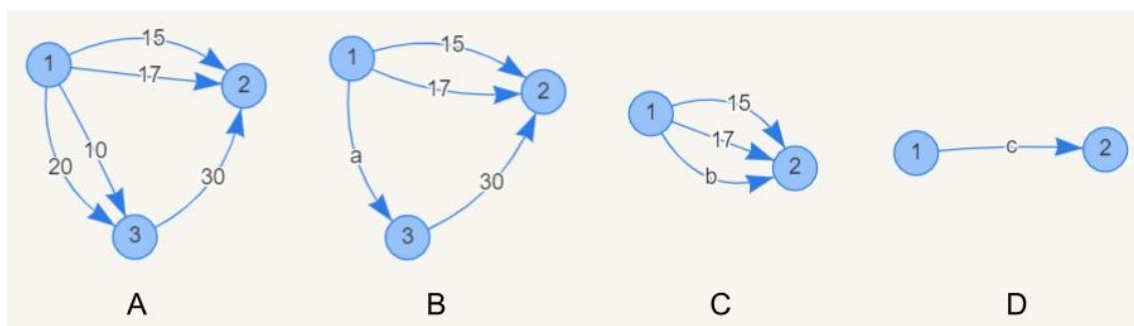


Figure 3: 範例說明

- A 中城鎮 1 至城鎮 3 的兩個傳送通道整體「蓋歐格」等於 B 中的 $a = \frac{1}{\frac{1}{10} + \frac{1}{20}} = \frac{20}{3}$
- B 中城鎮 1 至城鎮 3 及城鎮 3 至城鎮 2 的兩個傳送通道整體「蓋歐格」等於 C 中的 $b = a + 30 = \frac{110}{3}$
- C 中城鎮 1 至城鎮 2 的三個傳送通道整體「蓋歐格」等於 D 中的 $c = \frac{1}{\frac{1}{15} + \frac{1}{17} + \frac{1}{b}} = \frac{5640}{857} \approx 6.546091015$

F. 最小生成數

Problem ID: MinSpanNum

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: 唐可可可在上課打瞌睡，出自動畫《Love Live! Superstar!!》

獨自一人從上海前來日本成為學園偶像的唐可可，在路上偶然聽到了同校同學澀谷香音的美妙歌聲，情不自禁的用中文向香音說到「太好聽了吧！妳唱歌真的好好聽啊，簡直就是天籟！我剛才聽到妳唱歌了。我們以後一起唱好不好？一起唱！一起做學園偶像！」

在這之後，兩人便一起以成為學園偶像為目標而努力著。而為了準備第一次登場，兩人都花了許多時間和力氣，使得可可在上爆仔老師的數學課時不小心睡著了！

爆仔老師也在這時非常壞心地詢問可可問題，但因為她是全年級第一（華人的優良教育），所以老師拿出了他最近從古代演算法書《力扣》上面想出的問題——最小生成數。

老師叫醒可可後，向她說道：「給定一序列 a_i 和一個數字 k ，定義這個數列的生成數為 $b = f(a, k)$ ，使得任意長度為 k 的連續子區間內， $a_i < a_j \iff b_i < b_j$ 且 $a_i = a_j \iff b_i = b_j$ ，而當 b 數列內皆為正整數且為字典序最小的生成數時，我們稱其為最小生成數（MSN，Minimum Spanning Number）！」

可可雖然受過九年國民義務教育，但是對演算法相當不在行，你因為她非常可愛，為了和她建立良好關係，所以決定幫助她！請你幫她找出最小生成數吧！

— 輸入 —

第一行有二個正整數 n, k

其中 n 表示陣列 a 的長度

k 的定義和題目相同。

第二行有 n 個整數，表示給定的 a_i 。

— 輸出 —

輸出最小生成數。

— 輸入範圍 —

- $1 \leq n \leq 2 \times 10^5$
- $\forall i \in [1, n], \|a_i\| \leq 10^9$

— 子任務 —

| 編號 | 分數 | 額外限制 |
|----|----|-------------------|
| 0 | 0 | 範例輸入輸出 |
| 1 | 4 | $n \leq 10^3$ |
| 2 | 1 | $k = 1$ |
| 3 | 4 | $k = n$ 、保證全部元素相異 |
| 4 | 4 | $k = n$ |
| 5 | 8 | $k = 2$ 、保證全部元素相異 |
| 6 | 6 | $k = 2$ |
| 7 | 5 | 保證序列已經排序、全部元素相異 |
| 8 | 3 | 保證序列已經排序 |
| 9 | 20 | 保證全部元素相異 |
| 10 | 45 | 無額外限制 |

— 範例輸入 1 —

5 3
-3 3 0 2 -3

— 範例輸出 1 —

1 4 2 3 1

— 範例說明 1 —

a 中長度為 3 的連續子區間有 $[-3, 3, 0]$, $[3, 0, 2]$, $[0, 2, -3]$
區間 $[0, 2, -3]$ 重新生成後字典序最小的解為 $[2, 3, 1]$
受此影響, $[3, 0, 2]$ 生成出字典序最小的解為 $[4, 2, 3]$
而 $[-3, 3, 0]$ 中的 -3 則可直接生成為 1 而得到答案

— 範例輸入 2 —

5 2
3 2 -2 -1 0

— 範例輸出 2 —

3 2 1 2 3

— 範例輸入 3 —

5 4
-3 -1 -3 -1 -2

— 範例輸出 3 —

1 3 1 3 2

— 範例輸入 4 —

10 2
7 9 1 7 9 3 3 1 5 5

— 範例輸出 4 —

1 2 1 2 3 1 1 1 2 2