



# 2025 建北電資 X IZCC 聯合演算法競賽

## 試題本

---

### 競賽規則

1. 競賽時間：2025/1/2 17:45 ~ 19:45，共 2 小時。
2. 本競賽將在 cms 上進行，允許的語言有 C++11/g++、C++14/g++、C++17/g++、C11/gcc
3. 賽中可以參考即時計分版，連結：<https://iscoj.fg.tp.edu.tw/cms/ranking>
4. 本次競賽試題共 7 題，編號為 A ~ G。
5. 為了愛護地球，本次競賽題本僅提供電子檔，不提供紙本。
6. 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。
7. 本次比賽出現同分時兩者並列，時間不納入排名標準
8. 本次比賽無限制 submission 數，非 AC 結果不會有任何懲罰。
9. 每題每次上傳間隔為 60 秒，裁判得視情況調整。
10. 賽中只得使用電腦教室提供的電腦及設備，不能使用自己的電腦
11. 比賽期間禁止使用通訊用具，請在比賽前調至靜音或關機（但可以帶耳機聽音樂）
12. 比賽期間不得以任何形式與其他參賽者討論、交談
13. 可以攜帶任意頁數的紙本資料，且可訪問簡報及其他網路上的參考資料
14. 賽中禁止使用任何 AI 工具
15. 若對題目有任何疑問可在 cms 的提問系統提問，或詢問現場學術。若題目有修正或 rejudge 等會在 Announcement 上更新相關訊息
16. 請不要以任何形式傷害現場設備、系統或學術，但可以嘴砲溫室楊偷約會不揪。
17. 如需使用 C++ 的 `std::cin` 或 `std::cout` 可將以下程式碼插入 main function 以及將 `endl` 取代為 `'\n'` 來優化輸入輸出速度。唯須注意不可與 `cstdio` 混用。

```
std::ios::sync_with_stdio(false);  
std::cin.tie(nullptr);
```

---

# Amicable Family

time limit per test: 1 second

memory limit per test: 256 megabytes

建電與北資自 1995 年合併，至今已傳承至 29 屆，並合稱建北電資，也因為如此，雙方也培養出了許多情感，或愛情，並且兩社合作密切，不只是春夏秋冬的大活動，連放學的小社課也都是一起舉辦的。

如果你覺得女校或男校的生活太無聊，建北電資會告訴你什麼叫把男女分校當成男女混校，我們常常一起出去社遊，也常常會聽到有北資的說，「要不要去建電社辦，諸如此類。在常人來看，建北電資根本就是同一個社團。

總而言之「建北電資是一個溫暖的大家庭」

有一天建電社長無語跟北資社長pomelo在聊天：

無語：欸你們這學期社員有多少啊？

Pomelo：不多，也就 73 個。

無語：太多了吧，我們這屆只有 52 個社員欸，這樣我們兩社總人數是多少啊， $73 + 52 \dots 125$  個！

小黑（不知道從哪邊跑出來）：但這樣沒辦法成雙成對欸，我不想變成孤兒嗚嗚嗚嗚嗚

.....

如果建北電資的社員總數為奇數，小黑就會成為那個孤兒。而這時無語把兩社的人數告訴了你，請寫出一個程式判斷小黑會不會成為孤兒。

## Input

無語會給你兩個數字  $a, b$  ( $0 \leq a, b < 10^{100\,000}$ ) — 建中電研與北一資研的人數。

## Output

如果小黑會成為孤兒，請輸出 Yes，否則輸出 No。

## Scoring

Group	Add. constraints	Points
1	範例測資	1
2	$n \leq 10^9$	22
3	$n \leq 5 \cdot 10^{18}$	40
4	$n \leq 10^{19}$	9
5	—	28

## Example

input
52 73
output
Yes

# Bachelor

time limit per test: 1 second

memory limit per test: 256 megabytes

*你真的很可悲欸 比我還可悲那種*

— keaucucal, 建電一五學術長

kea 鄙視一切放閃行為，但身為單身漢的 kea 內心其實也相當羨慕有對象可以放閃，於是他決定踏上相親之路。

已知有  $n$  個相親對象，每個對象都會給他一個日期  $d$ ，在那天前 (不含當天) 都可以接受約出來相親，否則逾期就去找別人了。

對於每個相親對象，kea 會根據自己有多喜歡她/他，評估出一個幸福指數  $r$ ，代表的是跟他相親，kea 會多幸福，因為 kea 平時沒人約，所以有無限天可以拿來相親。

kea 想要盡可能的幸福，也就是希望能獲得越多幸福指數越好。但他不是時間管理大師，一天只能跟一個人相親，請問最大幸福指數是多少呢？

## Input

第一行包含一個正整數  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — 有幾個相親對象。

接下來有  $n$  行，每行包含兩個正整數  $d_i, r_i$  — 相親期限與幸福指數。

## Output

輸出一個正整數：能獲得的最大幸福指數。

## Scoring

Group	Add. constraints	Points
1	範例測資	2
2	$\forall i, j (1 \leq i < j \leq n) \implies d_i \neq d_j$	8
3	$\forall i, j (1 \leq i < j \leq n) \implies r_i = r_j$	12
4	$\forall i, j (1 \leq i, j \leq n) \quad d_i < d_j \implies r_i > r_j$	18
5	—	60

## Example

input
5 2 70 1 30 1 50 2 80 1 10
output
150

## Note

對於範例測資：

第一天和第一個對象相親，獲得 70 點幸福指數，第二天與第四個對象相親，獲得 80 點幸福指數，總共  $70 + 80 = 150$  點幸福指數

# Capitalism

time limit per test: 1 second

memory limit per test: 512 megabytes

在資本主義社會中，資本的運作是經濟活動的核心。股票市場是資本主義體系中一個極為重要的組成部分，投資者利用股票交易來增值自己的資本，並且為企業提供資金來促進發展。在這個過程中，如何抓住市場波動中的投資機會，最大化利潤，成為了每一位投資者都在思考的問題。

小明最近對股票市場產生了濃厚的興趣，他決定進行一次模擬投資，並且希望從中獲取最大的利潤。根據市場的變化，他可以在一段給定的時間內選擇進行一次買入和一次賣出的操作。每當股票的價格波動時，小明需要決定什麼時候買入，什麼時候賣出，以最大化他的投資回報。

然而，這次模擬投資有一個挑戰 — 小明「必須且只能進行一次買入和一次賣出」，而他需要在一個確定的時間段內選擇這兩個操作。最重要的是，賠錢也是可能的，因為有時候市場的變化對他來說並不友好。那麼，在給定的時間段內，最大利潤是多少呢？

你可以在同個時間點買入賣出，但仍需要負擔該點的漲跌，舉例而言，如果有一個時間段是  $-1, -1, -1, -1$ ，你選擇在其中一點買入賣出，則需要負擔  $-1$  元的漲跌。

## Input

第一行包含一個正整數  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — 時間段的長度。

第二行有  $n$  個整數  $x_i$  ( $-10^9 \leq x_i \leq 10^9$ ) — 每個時間點的漲跌幅度。

## Output

輸出一個整數 — 最大利潤。

## Scoring

Group	Add. constraints	Points
1	範例測資	0
2	$\forall i (1 \leq i \leq n) \implies x_i \geq 0$	4
3	$\forall i (1 \leq i \leq n) \implies x_i \leq 0$	7
4	$\exists i (1 \leq i \leq n) (\forall j < i (x_j < 0) \wedge \forall j \geq i (x_j \geq 0))$	12
5	$\exists i (1 \leq i \leq n) (\forall j < i (x_j \geq 0) \wedge \forall j \geq i (x_j < 0))$	12
6	—	65

## Example

input
8 -1 3 -2 5 3 -5 2 2
output
9

## Note

對於範例測資：

在第二個時間點買入並且在第四個時間點賣出，利潤  $= 3 - 2 + 5 + 3 = 9$  為最佳解

# Diligent Mr. Tsai

time limit per test: 1 second

memory limit per test: 256 megabytes

蔡哥非常邊緣，班上都沒有人揪他去看跨年煙火，所以他只好自己前往 101 看煙火。到了象山站後，場景只能說是 people mountain people sea。傻逼的他，完全找不到路。這時他突然想起自己化學老師的作業中寫過接近中心性的報告。

接近中心性 — 是指一個節點在網路中的緊密程度，可預期一個節點如果距離其他所有節點越近，它理當會處於較為中心的位置。計算方式非常簡單：該節點到其他所有節點的最短路徑長度取平均值。因此，他決定利用這個概念來找到關鍵的地點，幫助自己順利看到煙火。

聰明的你，請幫助蔡哥寫一個程式計算吧。另外，因為蔡哥最近忘記除法運算怎麼做了。因此，你只需要輸出每個點到所有其他點的最短路徑長度總和即可。

## Input

第一行有兩個正整數  $n, m$  ( $1 \leq n \leq 3000, 1 \leq m \leq 5000$ ) — 節點和邊的數量。

接下來有  $m$  行，每行包含兩個整數  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ) — 有一條雙向邊連結  $u_i, v_i$  兩點。

保證輸入的圖是一張沒有自環、重邊的簡單連通圖。

## Output

輸出  $n$  個數字，以空格隔開，分別為每個節點到其他所有節點的最短路徑長度和，數字之間以空格隔開。

## Scoring

Group	Add. constraints	Points
1	範例測資	2
2	$1 \leq n \leq 50, m \leq 100$	18
3	$1 \leq n \leq 500, m \leq 1000$	30
4	—	50

## Example

input
5 8 3 4 1 2 4 5 2 3 3 5 1 3 1 5 2 4
output
5 5 4 5 5

# Easy choice

time limit per test: 1 second

memory limit per test: 256 megabytes

存在  $N$  個物品，編號由 1 到  $N$ ，編號  $i$  的物品有其價值  $V_i$  及代價  $C_i$ ，選擇其中一些物品。

假設  $m$  為所選物品的數量而  $(A_1, A_2, \dots, A_m)$  為所選物品的編號集合 ( $\forall 1 \leq i < j \leq m, A_i < A_j$ )。

必須滿足  $\forall 1 \leq i < j \leq m, A_j - A_i \leq K, A_1 = 1, A_m = n$ 。

對於選取物品的分數定義為  $\left\lfloor \frac{\sum_{i=1}^m V_{A_i}}{\sum_{i=1}^m C_{A_i}} \right\rfloor$ ，請求出滿足條件下所選物品分數的最大值

## Input

第一行包含兩個整數  $N, K$  ( $1 \leq K \leq N \leq 2 \cdot 10^5$ )

第二行包含  $N$  個整數，第  $i$  個整數代表  $V_i$  ( $1 \leq V_i \leq 10^5$ )

第三行包含  $N$  個整數，第  $i$  個整數代表  $C_i$  ( $1 \leq C_i \leq 10^5$ )

## Output

輸出一個整數表示答案

## Scoring

Group	Add. constraints	Points
1	$1 \leq N \leq 3000$	17
2	—	83

## Examples

<b>input</b>
2 1 3 3 1 2
<b>output</b>
2

<b>input</b>
5 3 1 2 3 4 5 1 1 1 1 1
<b>output</b>
3



# Finite-State Machine

time limit per test: 1.5 seconds

memory limit per test: 256 megabytes

不是啊 她又不喜歡我 我也不喜歡她  
— 溫室羊, 建資百十五學術

「不要靠邀 一週內暈船我好強」一週後他如此說道，正如同溫室羊的暈船轉換過程，有限狀態機（Finite-State Machine，簡稱 FSM）是 腦癱大學(NauTan University) Emo Everyday 學系（簡稱 NTUEE）大一必修課之一 交換電路與邏輯設計 的重點概念之一。其中一種常見的有限狀態機稱之為 Mealy Machine，其特點是輸出由當前 State 和當前輸入共同決定。每個狀態會根據當前的輸入選擇對應的狀態轉移，同時生成相應的輸出。因此，輸出是即時更新的，隨著輸入的改變而變化。

一個 Mealy Machine 可以用一張 State Table 來表示。如下圖所示，表中一橫列代表一個 State，當機器處於某 Present State 且接收到輸入  $x$  時，將生成輸出  $z$ ，並在下一個時間點轉移到 Next State。

<i>Present</i> <i>State</i>	<i>Next state</i>		<i>Output (z)</i>	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
$S_0$	$S_3$	$S_2$	0	1
$S_1$	$S_8$	$S_7$	0	1
$S_2$	$S_4$	$S_3$	1	0
$S_3$	$S_0$	$S_4$	0	1
$S_4$	$S_2$	$S_0$	1	0
$S_5$	$S_5$	$S_1$	1	0
$S_6$	$S_1$	$S_7$	0	1
$S_7$	$S_2$	$S_6$	1	0
$S_8$	$S_5$	$S_1$	1	0

State table.

我們可以注意到，若一個有限狀態機有  $N$  個狀態（State），則至少需要  $\lceil \log_2 N \rceil$  個位元來表示每個狀態。然而，在 CPU 中儲存每個位元所需的電晶體成本相當高昂，因此我們需要對狀態進行化簡以降低資源消耗。

我們將兩個狀態定義為「等價」，若且唯若它們在所有可能的輸入（Input）下具有相同的輸出（Output），且其對應的下一狀態（Next State）也指向相同的等價狀態。例如，上表中的  $S_5$  和  $S_8$  即為等價狀態。

化簡狀態的第一步驟是識別所有等價的狀態，並將它們合併為單一狀態，如下表所示，從而減少狀態數量，提升有限狀態機的效率。

Present State	Next state		Output (z)	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
$S_0$	$S_3$	$S_2$	0	1
$S_1$	<del><math>S_8</math></del> $S_5$	$S_7$	0	1
$S_2$	$S_4$	$S_3$	1	0
$S_3$	$S_0$	$S_4$	0	1
$S_4$	$S_2$	$S_0$	1	0
$S_5$	$S_5$	$S_1$	1	0
$S_6$	$S_1$	$S_7$	0	1
$S_7$	$S_2$	$S_6$	1	0
<del><math>S_8</math></del>	<del><math>S_5</math></del>	<del><math>S_1</math></del>	<del>1</del>	<del>0</del>

同時，在上表當中，我們可以注意到， $S_0 = S_3$  若且唯若  $S_3 = S_0$  且  $S_2 = S_4$ ， $S_2 = S_4$  若且唯若  $S_4 = S_2$  且  $S_3 = S_0$ ，因為這兩個條件句無矛盾，所以  $S_0 = S_3$  和  $S_2 = S_4$  同時成立。因此，這張表最後可以化簡為 6 個 State。

上述做法可以使用稱之為 Implication Table 的方式完成。如下圖所示，先將每個 State 要相同的條件寫在表格中，並且將絕對不可能的情況打叉（例如 Output 不同的兩個 State）。之後，不斷地檢查每一個條件是否已經無法達成了，若無法達成就打叉，直到表中每一格都無法被打叉為止，剩下每一個沒被打叉的格子就代表一組 State 相等。

$S_1$	3-5 2-7						
$S_2$	×	×					
$S_3$	2-4	0-5 4-7	×				
$S_4$	×	×	0-3	×			
$S_5$	×	×	1-3 4-5	×	2-5 0-1		
$S_6$	1-3 2-7	1-5	×	0-1 4-7	×	×	
$S_7$	×	×	2-4 3-6	×	0-6	2-5 1-6	×
	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$

Implication table.

上面這一坨化簡的方式，Aaron Wu 完全都看不懂。因此，他打算找另一個 Aaron Wu 幫忙，請他寫一個程式，只要輸入一整張 State Table，就會輸出哪些 State 相等。然而，另一個 Aaron Wu 忙著在處理脆上的公關危機，因此，你能幫助 Aaron Wu 們完成這份程式嗎？

Input

第一行有兩個正整數  $n, m$  ( $2 \leq n \leq 1000, 2 \leq m \leq 16$ ) — State 的數量和可能的 Output 數。

接下來  $n$  行，每行有  $m$  個整數  $q_{ij}$  ( $0 \leq q_{ij} < n$ ) — 當狀態為  $S_i$ ，input 為  $j$  時，會在下一個時間轉為  $S_{q_{ij}}$

接下來  $n$  行每行有  $m$  個整數  $z_{ij}$  ( $0 \leq z_{ij} \leq 1$ ) — 當狀態為  $S_i$ ，input 為  $j$  時，FSM 會 Output  $z_{ij}$

Output

請輸出若干行，每行兩個整數  $a < b$ ，代表  $S_a = S_b$ 。

若無任何 State 相等，請輸出 Emo Everyday

另外，請排序後再輸出（詳見範例1）

Scoring

Group	Add. constraints	Points
1	$n \leq 10$	14
2	保證不等價的 state 都有不同的 output	8
3	—	78

Examples

input
5 2 1 4 1 4 2 0 1 4 2 0 0 0 0 0 1 0 0 0 1 0
output
0 1 0 3 1 3 2 4

input
9 2 3 2 8 7 4 3 0 4 2 0 5 1 1 7 2 6 5 1

0 1
0 1
1 0
0 1
1 0
1 0
0 1
1 0
1 0
output
0 3
2 4
5 8

input
2 4
0 1 0 1
1 0 1 0
0 0 0 1
1 1 0 1
output
Emo Everyday

Note

說明文章：<https://reurl.cc/74VX0y>

對於 Sample 1，State Table 如下所示，可見  $S_0 = S_1 = S_3, S_2 = S_4$

<i>Present State</i>	<i>Next state</i>		<i>Output (z)</i>	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
$S_0$	$S_1$	$S_4$	0	0
$S_1$	$S_1$	$S_4$	0	0
$S_2$	$S_2$	$S_0$	1	0
$S_3$	$S_1$	$S_4$	0	0
$S_4$	$S_2$	$S_0$	1	0

對於 Sample 3，State Table 如下所示，可見  $S_0$  和  $S_1$  Output 不同，因此無法化簡

<i>Present State</i>	<i>Next state</i>				<i>Output (z)</i>			
	$x = 0$	$x = 1$	$x = 2$	$x = 3$	$x = 0$	$x = 1$	$x = 2$	$x = 3$
$S_0$	$S_0$	$S_1$	$S_0$	$S_1$	0	0	0	1
$S_1$	$S_1$	$S_0$	$S_1$	$S_0$	1	1	0	1

# Gulf

time limit per test: 1 second  
memory limit per test: 256 megabytes

暈船是什麼可怕的東西 不要暈船

— Ian Wen, 建電一五網管

是的，Ian Wen 不會暈船，他只知道如何讓別人暈船，是個名副其實的大海王。他的休閒日常就是在海灣釣魚，且他作為一個王，想當然他也有一個碩大的皇宮，而皇宮的後院，簡稱後宮，則有  $n$  個魚塘，裡面養著各種 Ian Wen 釣過的魚，而這些魚如果長時間沒有被 Ian Wen 滋潤到，他們會逐漸喪失活力，更甚者離開魚塘，這是 Ian Wen 最不樂見的。為了避免這種情況發生，他決定在後宮建一些雨水採集器和管道讓他可以持續的滋潤水塘。

對於每一個魚塘，在正上方建立一個雨水採集器的價錢為  $c_i$  ( $1 \leq i \leq n$ )，而每一個採集器可以接無限多個管道，但每個管道只能從一個魚塘上方出來滋潤該魚塘。若此管道是從位置為  $s$  的採集器出發，且出口在  $k$ ，則建立此管道的價格為  $|k - s|$ 。

Ian Wen 希望自己可以顧到所有  $n$  個魚塘，所以他決定在每個魚塘上面都建立一個管道，但因為他的金力有限，所以他想知道最少要花多少錢才可以讓每隻魚不喪失活力且不離開他。

## Input

第一行包含一個整數  $n$  ( $1 \leq n \leq 5000$ ) — 魚塘的數量。

第二行有  $n$  個整數  $c_i$  ( $1 \leq c_i \leq 10^9$ ) — 在此魚塘上方建立雨水採集器需要的花費。

## Output

輸出一個整數 — 需要花多少錢才可以讓每一個魚塘上方都至少有一個管道。

## Scoring

Group	Add. constraints	Points
1	$n \leq 20$	20
2	$n \leq 500$	30
3	—	50

## Example

input
6 7 1 8 6 8 2
output
8

## Note

對於第一筆測資：

在編號 2 和 6 的魚塘蓋雨水採集器，價格分別為 1 和 2，並且從魚塘 2 建立四個管道分別連接編號 1 ~ 4 的魚塘，從 6 建立兩個管道連結 5 和 6 的魚塘。

成本為  $1 + 2 + |1 - 2| + |2 - 2| + |3 - 2| + |4 - 2| + |5 - 6| + |6 - 6| = 8$ 。此解為最小花費。