

Contest Name : Day 5 總召好帥之總召真的好帥正式賽

Starts at	Ends at	Contest Type
2021-01-31 10:10:00	2021-01-31 15:10:00	acm style (no partial/dashboard)

Description

注意事項

1. 請使用團體帳號參與此競賽。
2. 競賽時間共 300 分鐘。**每一隊每 5 秒只能上傳一次程式碼**。本次比賽將開放計分板，位於 Dashboard 的地方。若於題目上有疑慮，亦可以於以上 Q&A 的地方提問。
3. 比賽會在最後 60 分鐘停止更新記分板。
4. 每一題的題目有6個部分，分別是題目敘述（Description）、輸入格式（Input Format）、輸出格式（Output Format）、範例輸入（Sample Input）、範例輸出（Sample Output）、提示（Hints），請仔細閱讀後再進行作答。
5. 本次競賽使用線上評測系統 TIOJ Infor Online Judge IOICamp edition（即本站）。參賽者需將程式碼上傳至線上評測系統，才予以計分。成績以評測系統上的分數為準。
（若發現測資有誤等情形，仍有可能在比賽中或結束後重新測試，且以重新測試後之分數作為最終分數。）
6. 本次比賽，可以用 C++14、C++17、C11 等語言作答。
7. 競賽的開始、結束時間以線上評測系統為準，只有在競賽期間上傳的程式碼才會計分。（以上傳程式碼的時間為準。若在競賽期間內上傳，但評測完畢時競賽已結束，仍然會計分。）
8. 若題目內容有修正，將會發佈公告，並以線上評測系統的題目內容為主。
9. 競賽期間，學員應隨時備份資料，以防資料流失。
10. 競賽中只允許開啟撰寫程式必需的編輯器、輔助程式（限記事本、小算盤、小畫家），或使用 Discord 與隊上成員交流，不得開啟其他無關的應用程式，也不得連結本站以外的其他網站，違者視同作弊。
11. 競賽期間，請勿交談或與其他隊的學員有任何交流，違者視同作弊。
12. 競賽期間，禁止攜帶或使用任何電子設備以及使用無線網路，違者視同作弊。
13. 競賽期間，禁止以任何方式攻擊學校網路或評測系統，違者視同作弊。
14. 由於線上評測系統採自動化評分，**請參賽者務必看清題目敘述（包含範例輸入、輸出與提示），並嚴格遵守各題的輸入輸出格式**。若格式錯誤，將可能被判斷為 **WA（答案錯誤）**，並獲得 0 分。
15. 若對評測得到的結果不甚了解，可參考**傳送結果介紹**。
16. 競賽禁止使用自備的鍵盤、滑鼠等配件，請使用競賽場所所提供的設備。
17. 總召好帥。
18. 上傳前如果沒有大喊一聲「總召好帥」讓全體參加人員聽到，該筆 submission 可能會被評為 **WA（答案錯誤）**。

glhf OwO/

Tasks

pA 53. 生成樹

pB 48. 如果我有一座新冰箱

pC 55. 帥哥總召算雜湊

pD 47. 火鍋吃到飽問題

pE 51. 卑鄙源之助

pF 54. 階梯序列

pG 56. 總召帥不帥

pH 52. 最短路徑生成樹

pI 45. BB 與 ㄌㄌ

pJ 50. 鬼轉影片

pK 46. IOIC 王國

pL 49. 時代的卷跡

pM 44. 黑洞危機

53. 生成樹

Submit

Status

Ranklist

Back to Problems List

TopCoder



ioicamp_47

User's AC Ratio

100.0% (1/1)

Submission's AC Ratio

100.0% (1/1)

Tags

Description

給你一張無向帶權連通圖。這張圖有 N 個點、 M 條邊。點編號為 1 到 N 。第 i 條邊連接著點 u_i 以及點 v_i ，並且邊權為 w_i 。

現在，你有 Q 個詢問。每個詢問會給你一條帶權邊 E 。請問最少要刪除多少條邊，才能讓 E 被加進圖後能夠出現在某一個最大權生成樹中，以及出現在某一個最小權生成樹中。

注意到，每一筆詢問是彼此獨立的。

Input Format

輸入的第一行包含兩個正整數 N, M ，代表這張圖的點數、邊數。

接下來的 M 行，第 i 行包含三個正整數 u_i, v_i, w_i ，代表第 i 條邊連接點 u_i 以及點 v_i ，並且邊權為 w_i 。

接下來的一行，包含一個正整數 Q ，代表詢問個數。

接下來的 Q 行，第 i 行包含三個正整數 u, v, w ，代表第 i 個詢問的邊，是連接點 u 、點 v ，並且邊權為 w 。

- $2 \leq N \leq 500$
- $N - 1 \leq M \leq 1000$
- $1 \leq u_i, v_i, u, v \leq N$
- $u_i \neq v_i, u \neq v$
- $1 \leq w_i, w \leq 1000$
- $1 \leq Q \leq 10$

Output Format

對於每筆詢問，請輸出一個整數，代表最少需要刪除多少邊，才能讓 E 被加進圖後能夠出現在任何一個最大權生成樹中，以及出現在任何一個最小權生成樹中。

Sample Input

copy

```
5 7
1 2 1
2 3 3
3 1 5
3 4 7
4 5 9
5 3 11
3 5 13
2
3 4 2
5 1 8
```

Sample Output

copy

```
2
0
```

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⤴
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	
7	1000	262144	65536	
8	1000	262144	65536	
9	1000	262144	65536	
10	1000	262144	65536	
11	1000	262144	65536	
12	1000	262144	65536	
13	1000	262144	65536	
14	1000	262144	65536	
15	1000	262144	65536	
16	1000	262144	65536	
17	1000	262144	65536	
18	1000	262144	65536	
19	1000	262144	65536	
20	1000	262144	65536	
21	1000	262144	65536	
22	1000	262144	65536	
23	1000	262144	65536	
24	1000	262144	65536	
25	1000	262144	65536	
26	1000	262144	65536	
27	1000	262144	65536	

48. 如果我有一座新冰箱


Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上，我弱

zck 至高無上

User's AC Ratio

100.0% (1/1)

Submission's AC Ratio

100.0% (1/1)

Tags

Description

身為一個資深ㄌㄌ控，bb 買了一座有 N 個格子的新冰箱來冷藏他的ㄌㄌ收藏，格子的編號依序由左往右是 $1 \sim N$ 。

但是 bb 收藏的ㄌㄌ都很傲嬌，她們討厭和其他ㄌㄌ分享 bb，因此希望離其他ㄌㄌ越遠越好。每當一個ㄌㄌ要進入冰箱，她會去找一個位置使得離他最近的ㄌㄌ距離盡量遠，如果有很多位置離最近的ㄌㄌ距離都相同，她會選擇其中最左邊的。而且因為這些ㄌㄌ都很傲嬌，她們也不希望離冰箱的邊緣太近，他們對冰箱邊緣的厭惡度等同於你在第 0 格和第 $N + 1$ 格都擺上一個ㄌㄌ。

作為專業的ㄌㄌ收藏家，bb 把每個ㄌㄌ都設置了一個屬性（屬性是一個 $1 \sim 10^9$ 的整數）。但 bb 是個喜新厭舊的人，他有時候會把某個屬性的ㄌㄌ全部趕出去，以空出空間容納新的ㄌㄌ。

bb 的好友小 y 和小 p 想要知道在 2020 年，bb 收（日ㄌㄌ）藏（ㄌㄌ）了多少蘿莉，於是就偷走了 bb 的記帳手冊。上面詳列了 bb 收養了哪些ㄌㄌ以及他趕走了哪些ㄌㄌ，為了向警察杯杯完整還原事發經過，小 y 和小 p 打算在記錄裡補上每一個ㄌㄌ被收養之後住進冰箱的哪個格子。

在認真整理之後，小 y 和小 p 發現 bb 的ㄌㄌ數量大到連他自己都搞不清楚，有時候他想要趕走的ㄌㄌ屬性根本不存在（這種時候當然什麼都沒趕走），有時候他把ㄌㄌ帶回冰箱的時候ㄌㄌ冰箱是滿的，這樣ㄌㄌ就會自己落跑。小 y 和小 p 已經處理到頭腦過熱想要放進冰箱冷卻一下了，請你寫一隻程式幫他們整理這些紀錄吧！

Input Format

第一行有兩個正整數 N, Q ，代表 bb 的新冰箱有 N 格。

第二行有 N 個整數 a_i ，依序代表在 2020 年開始之前，住在第 i 個格子的ㄌㄌ的屬性，如果 $a_i = 0$ 代表第 i 個格子沒有住ㄌㄌ。

接下來的 Q 行，每行有兩個正整數 t_i, k_i ，代表第 i 筆紀錄的內容。如果 $t_i = 0$ ，表示 bb 把一隻屬性是 k_i 的ㄌㄌ加入冰箱裡。如果 $t_i = 1$ ，表示 bb 把所有屬性是 k_i 的ㄌㄌ都趕出冰箱了。

- $1 \leq N, Q \leq 2 \times 10^5$
- $0 \leq a_i \leq 10^9$
- $t_i \in \{0, 1\}$
- $1 \leq k_i \leq 10^9$

Output Format

對於每一筆 $t_i = 0$ 的紀錄輸出一行，包含一個正整數 pos ，代表這隻ㄌㄌ入住了冰箱的第 pos 格。

如果這個時候冰箱根本是滿的，請輸出 **-1** 並不要更動冰箱的內容。

Sample Input copy

```
5 6
0 10 3 0 2
1 10
0 3
0 9
1 3
1 2
0 8
```

Sample Output copy

```
1
2
4
```

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	3000	262144	65536	
1	3000	262144	65536	
2	3000	262144	65536	
3	3000	262144	65536	
4	3000	262144	65536	
5	3000	262144	65536	
6	3000	262144	65536	
7	3000	262144	65536	
8	3000	262144	65536	
9	3000	262144	65536	
10	3000	262144	65536	
11	3000	262144	65536	
12	3000	262144	65536	
13	3000	262144	65536	
14	3000	262144	65536	
15	3000	262144	65536	
16	3000	262144	65536	
17	3000	262144	65536	
18	3000	262144	65536	
19	3000	262144	65536	
20	3000	262144	65536	
21	3000	262144	65536	
22	3000	262144	65536	
23	3000	262144	65536	
24	3000	262144	65536	
25	3000	262144	65536	
26	3000	262144	65536	
27	3000	262144	65536	
28	3000	262144	65536	
29	3000	262144	65536	

— 55. 帥哥總召算雜湊

Submit

Status

Ranklist

Back to Problems List

TopCoder

User's AC Ratio

Submission's AC Ratio

Tags

NaN% (0/0)

NaN% (0/0)

Description

今天帥哥總召在做字串問題時，突然看到了一個字串雜湊演算法，此演算法的實作方法如下：

```
int hash(string s, int n, int m, int p) {
    int h = 0;
    for (int i = 0; i < n; i++) h = (h * p + s[i]) % m;
    return h;
}
```

其中 s 是要拿來做雜湊的字串， n 是字串 s 的長度，而 m, p 皆為雜湊函數的參數且滿足 m 是質數。聰明的帥哥總召當然知道雜湊函數是一種多對一的函數因此有可能會發生碰撞，因此他開始在想，在給定 n, m, p 的情況下，若考慮所有長度為 n 僅包含大寫英文字母 (A-Z) 的字串，從中挑出任兩個相異的字串發生碰撞的機率為何？兩字串 s, t 發生碰撞代表 $hash(s, n, m, p) = hash(t, n, m, p)$ ，由於這個問題太過簡單不需要勞煩帥哥總召親自出馬，請你們趕緊幫他算出這個問題的答案。

Input Format

Output Format

輸入僅包含一行三個正整數 n, m, p 如題目所述。

- $1 \leq n \leq 10^6$
- $2 \leq m, p \leq 3 \times 10^4$
- 保證 m 是質數

請求出題目所述問題的答案。已知答案可以表示成最簡分數 $\frac{P}{Q}$ ，請輸出 $P \cdot Q^{-1} \bmod 998243$ 的值。

Sample Input

copy

Sample Input #1
1 3 2

Sample Input #2
2 3 2

Sample Input #3
12 13 16

Sample Input #4
94513 8737 9024

Sample Input #5
864303 27791 28661

Sample Output

copy

Sample Output #1
307152

Sample Output #2
869279

Sample Output #3
366275

Sample Output #4
280600

Sample Output #5
892581

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	3000	262144	65536	
1	3000	262144	65536	
2	3000	262144	65536	
3	3000	262144	65536	
4	3000	262144	65536	
5	3000	262144	65536	
6	3000	262144	65536	
7	3000	262144	65536	
8	3000	262144	65536	
9	3000	262144	65536	
10	3000	262144	65536	
11	3000	262144	65536	
12	3000	262144	65536	
13	3000	262144	65536	
14	3000	262144	65536	
15	3000	262144	65536	
16	3000	262144	65536	
17	3000	262144	65536	
18	3000	262144	65536	
19	3000	262144	65536	
20	3000	262144	65536	
21	3000	262144	65536	
22	3000	262144	65536	
23	3000	262144	65536	
24	3000	262144	65536	
25	3000	262144	65536	
26	3000	262144	65536	
27	3000	262144	65536	
28	3000	262144	65536	
29	3000	262144	65536	

Submit

Status

Ranklist

Back to Top

— 47. 火鍋吃到飽問題

Submit

Status

Ranklist

Back to Problems List

TopCoder

User's AC Ratio

Submission's AC Ratio

Tags

NaN% (0/0)

NaN% (0/0)

Description

作為一個精打細算的消費者，每當到了吃到飽餐廳消費一定要吃到撐為止。

今天你的朋友來到了一間火鍋吃到飽，然而身為大胃王的他發現食量的瓶頸在於餐廳限時而不是他的胃！

我們已知餐廳限時 T 分鐘，無限量供應 N 種食材，且食材 i ：

- 每單位需要煮 c_i 分鐘。由於火鍋容量很小，因此我們假定同時只能煮一單位的食物。
- 煮熟後的 A 分鐘內為最佳賞味時間。為了避免食物冷掉，你的朋友預計在最佳賞味期限內花 e_i 單位時間吃完一單位的食物。
- 每單位食材 i 能夠提供的滿足度為 s_i 。
- 如果 $i > j$ ，則食材 i 一定比食材 j 更重口味且容易弄髒湯底。已知一旦煮了重口味的食材弄髒湯底就不可挽回了！而且你的朋友喜歡吃食物的原味，希望盡量保持湯底清澈，所以越重口味的食材要放在越後面的順位烹煮。

給定以上資訊，請你幫他規劃一個煮火鍋的行程，使得他能夠獲得的最大滿足度。

Input Format

第一行有 3 個整數 T 、 N 和 A ，分別為餐廳限時、食材數量和最佳賞味時間。

接下來 N 行，第 $i + 1$ 行有 3 個整數 c_i 、 e_i 和 s_i ，分別為煮一單位食物時間、吃一單位食物的時間和滿足度。

- $1 \leq T \leq 1000$
- $1 \leq N \leq 1000$
- $1 \leq A \leq 10$
- $1 \leq c_i \leq T$
- $1 \leq e_i \leq A$
- $1 \leq s_i \leq 10^5$

Output Format

請輸出一個整數，代表根據你所安排的煮火鍋順序，能夠獲得的最大滿足度。

Sample Input

copy

```
Sample Input #1

20 3 4
8 4 4
3 2 3
4 1 2

Sample Input #2

19 3 4
8 4 4
2 4 3
4 1 2
```

Sample Output

copy

```
Sample Output #1

18

Sample Output #2

14
```

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	
7	1000	262144	65536	
8	1000	262144	65536	
9	1000	262144	65536	
10	1000	262144	65536	
11	1000	262144	65536	
12	1000	262144	65536	
13	1000	262144	65536	
14	1000	262144	65536	
15	1000	262144	65536	
16	1000	262144	65536	
17	1000	262144	65536	
18	1000	262144	65536	
19	1000	262144	65536	
20	1000	262144	65536	
21	1000	262144	65536	
22	1000	262144	65536	
23	1000	262144	65536	
24	1000	262144	65536	
25	1000	262144	65536	
26	1000	262144	65536	
27	1000	262144	65536	
28	1000	262144	65536	
29	1000	262144	65536	
30	1000	262144	65536	
31	1000	262144	65536	
32	1000	262144	65536	
33	1000	262144	65536	
34	1000	262144	65536	
35	1000	262144	65536	
36	1000	262144	65536	
37	1000	262144	65536	
38	1000	262144	65536	
39	1000	262144	65536	

51. 卑鄙源之助


Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上，我弱

zck 至高無上

User's AC Ratio

100.0% (1/1)

Submission's AC Ratio

100.0% (1/1)

Tags

Description

卑鄙源之助跟朋友在玩 nim，盤面已經決定好了，總共有 n 堆，第 i 堆有 a_i 個石子。

因為卑鄙源之助很卑鄙，他會偷藏薯條，也會偷藏石子，他可以遊戲開始前選其中幾堆藏起來。

因為卑鄙源之助很卑鄙，比賽過程中他也會作弊，他在比賽過程中，當輪到他時，他可以選擇做正常的 nim 操作，或是也可以選擇做卑鄙操作，以下是卑鄙操作的定義。

- 把場上其中一堆石子丟掉（那堆石子就不見了），然後把自己當初藏起來的一堆石子加入場上（加入的那些石子自成一堆，而且只能選擇當初藏起來的其中一堆石子，不能把多堆合起來或是只放一堆的其中一部份）。

如果卑鄙源之助做完操作之後，場面看起來完全沒有變化，對手會發現在作弊，因此如果進行卑鄙操作，丟掉的堆跟換上去的堆不能一樣大。

卑鄙源之助是後手，請問卑鄙源之助開始時有幾種藏石子的方法（幾種石子堆的子集合），可以使他獲勝。

註：

- 不藏或是全部都藏都是合法的
- 藏起不同編號的石子堆即算不同種藏法

Input Format

輸入第一行有一個正整數 n ，表示堆數 第二行有 n 個數字 $a_1, a_2, \cdots a_n$ ， a_i 表示第 i 堆的石子數。

- $1 \leq n \leq 26$
- $1 \leq a_i \leq 10^9$

Output Format

輸出一個數字，表示有幾種藏石子的方法，使卑鄙源之助獲勝。

Sample Input

copy

Sample Input #1

1
18

Sample Input #2

3
2 2 2

Sample Output

copy

Sample Output #1

1

Sample Output #2

4

Hints

Problem Source

Subtasks		
No.	Testdata Range	Score

Testdata and Limits					⌵
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1000	262144	65536		
1	1000	262144	65536		
2	1000	262144	65536		
3	1000	262144	65536		
4	1000	262144	65536		
5	1000	262144	65536		
6	1000	262144	65536		
7	1000	262144	65536		
8	1000	262144	65536		
9	1000	262144	65536		
10	1000	262144	65536		
11	1000	262144	65536		
12	1000	262144	65536		
13	1000	262144	65536		
14	1000	262144	65536		
15	1000	262144	65536		
16	1000	262144	65536		
17	1000	262144	65536		
18	1000	262144	65536		
19	1000	262144	65536		
20	1000	262144	65536		
21	1000	262144	65536		
22	1000	262144	65536		
23	1000	262144	65536		
24	1000	262144	65536		
25	1000	262144	65536		
26	1000	262144	65536		
27	1000	262144	65536		
28	1000	262144	65536		
29	1000	262144	65536		
30	1000	262144	65536		
31	1000	262144	65536		
32	1000	262144	65536		
33	1000	262144	65536		
34	1000	262144	65536		
35	1000	262144	65536		
36	1000	262144	65536		
37	1000	262144	65536		
38	1000	262144	65536		
39	1000	262144	65536		
40	1000	262144	65536		

✓ 54. 階梯序列


Submit

Status

Ranklist

Back to Problems List

TopCoder

 hhhhaura

User's AC Ratio

75.0% (3/4)

Submission's AC Ratio

25.0% (3/12)

Tags

Description

對一個序列 a_1, a_2, \dots, a_n ，找出最長的子序列 $a_{d_1}, a_{d_2}, \dots, a_{d_k}$ 滿足

- $a_{d_i} = i, \quad \forall 1 \leq i \leq k$
- $\gcd(d_i, d_{i+1}) > 1, \quad \forall 1 \leq i < k.$

Input Format

輸入第一行是一個正整數 T 代表測資筆數，在每筆測資中會先有一行一個正整數 n ，接下來是一行 n 個正整數 a_i 。

- $1 \leq a_i \leq n$
- 保證所有 n 總合不超過 3×10^5 。

Output Format

對每筆測資，輸出最長可能的長度。

Sample Input

copy

```
4
4
1 2 3 4
4
2 2 3 4
5
1 1 2 2 3
7
1 1 2 2 4 3 4
```

Sample Output

copy

```
1
0
2
3
```

Hints

Problem Source

Subtasks		
No.	Testdata Range	Score

Testdata and Limits					⌆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1000	262144	65536		
1	1000	262144	65536		
2	1000	262144	65536		
3	1000	262144	65536		
4	1000	262144	65536		
5	1000	262144	65536		
6	1000	262144	65536		
7	1000	262144	65536		
8	1000	262144	65536		
9	1000	262144	65536		
10	1000	262144	65536		
11	1000	262144	65536		
12	1000	262144	65536		
13	1000	262144	65536		
14	1000	262144	65536		

✓56. 總召帥不帥

Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上，我弱

zck 至高無上

User's AC Ratio

100.0% (1/1)

Submission's AC Ratio

50.0% (1/2)

Tags

Description

某英國研究指出，每個地球人都可以用 N 種特徵來描述，而這些特徵 XOR 起來可以判斷一個人的帥度：XOR 起來等於零代表這個人很帥，反之則不帥。現在給你總召的特徵 A_1, A_2, \dots, A_N ，請你判斷總召帥不帥。

Input Format

輸入的第一行有一個正整數 N 。

第二行有 N 個整數 A_1, A_2, \dots, A_N 。

- $1 \leq N \leq 10^5$
- $0 \leq A_i \leq 10^{18}$

Output Format

若總召很帥，請輸出 "YessssssssssS"；否則輸出 "Nooooooooooooo0"（不含引號）。

Sample Input

copy

Sample Input #1

1
1

Sample Input #2

2
1 1

Sample Input #3

3
1 2 3

Sample Output

copy

Sample Output #1

Nooooooooooooo0

Sample Output #2

YessssssssssS

Sample Output #3

YessssssssssS

Hints

Problem Source

Subtasks		
No.	Testdata Range	Score

Testdata and Limits					⌵
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1000	262144	65536		
1	1000	262144	65536		
2	1000	262144	65536		
3	1000	262144	65536		
4	1000	262144	65536		
5	1000	262144	65536		
6	1000	262144	65536		
7	1000	262144	65536		
8	1000	262144	65536		
9	1000	262144	65536		
10	1000	262144	65536		
11	1000	262144	65536		
12	1000	262144	65536		
13	1000	262144	65536		
14	1000	262144	65536		
15	1000	262144	65536		
16	1000	262144	65536		
17	1000	262144	65536		
18	1000	262144	65536		
19	1000	262144	65536		
20	1000	262144	65536		
21	1000	262144	65536		
22	1000	262144	65536		
23	1000	262144	65536		
24	1000	262144	65536		

52. 最短路徑生成樹

Submit

Status

Ranklist

Back to Problems List

TopCoder

User's AC Ratio

Submission's AC Ratio

Tags

NaN% (0/0)

NaN% (0/0)

Description

給出一個 N 點 M 邊的連通無向圖，任兩條邊的邊權皆不相同。

首先我們先從圖 G 上挑選出一個生成樹 T 。

接著對於一個點 u 來說，我們稱點 u 是「好點」若且唯若 u 想要以最短路徑抵達圖 G 上任一點都可以只透過這棵生成樹達到。

請找到一個滿足邊權總和最小的生成樹 T' ，倘若有多個可能請選擇有最多「好點」的那棵生成樹，如果仍有很多棵樹，則可以任選一棵，並找出這個樹上有哪些好點。

Input Format

Output Format

測試資料第一行包含兩個正整數 N, M ，代表圖 G 上的點數與邊數。

接下來一共有 M 行，每行會有三個正整數 u_i, v_i, w_i 代表圖上有一條 u_i 到 v_i 邊權為 w_i 的邊。

- $1 \leq N \leq 3 \times 10^5$
- $1 \leq M \leq 3 \times 10^5$
- $1 \leq u_i, v_i \leq N$
- $1 \leq w_i \leq 10^{12}$

第一行請輸出一個數字 n 代表你選擇的生成樹 T' 上有幾個「好點」。第二行請由小到大輸出一共 n 個數字，代表生成樹 T' 上每個好點的編號。

Sample Input

copy

8 8
1 2 100
2 3 20
2 8 19
8 6 18
6 7 23
3 4 10
4 5 2
4 7 21

Sample Output

copy

2
1 2

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	2000	262144	65536	
1	2000	262144	65536	
2	2000	262144	65536	
3	2000	262144	65536	
4	2000	262144	65536	
5	2000	262144	65536	
6	2000	262144	65536	
7	2000	262144	65536	
8	2000	262144	65536	
9	2000	262144	65536	
10	2000	262144	65536	
11	2000	262144	65536	
12	2000	262144	65536	
13	2000	262144	65536	
14	2000	262144	65536	
15	2000	262144	65536	
16	2000	262144	65536	
17	2000	262144	65536	
18	2000	262144	65536	
19	2000	262144	65536	
20	2000	262144	65536	
21	2000	262144	65536	
22	2000	262144	65536	
23	2000	262144	65536	
24	2000	262144	65536	
25	2000	262144	65536	
26	2000	262144	65536	
27	2000	262144	65536	
28	2000	262144	65536	
29	2000	262144	65536	
30	2000	262144	65536	
31	2000	262144	65536	
32	2000	262144	65536	
33	2000	262144	65536	

45. BB 與 ㄌㄌ


Submit

Status

Ranklist

Back to Problems List

TopCoder



team_12

User's AC Ratio

100.0% (1/1)

Submission's AC Ratio

100.0% (1/1)

Tags

Description

BB 是個ㄌㄌ控。他養了 6 個ㄌㄌ，編號為 1 到 6。

BB 每天都會跟恰好一個ㄌㄌ玩耍。他每天會丟一個公正的六面骰，來決定他要跟哪個ㄌㄌ玩。

而 BB 對一些ㄌㄌ的組合有著特別的喜好。他總共對於 N 個ㄌㄌ組合特別有興趣，第 i 個ㄌㄌ組合會以一個由字元 '1' 到字元 '6' 所構成的字串 s_i 來表示。代表說如果 BB 連續依序跟那些ㄌㄌ玩，他會非常爽，並且馬上陷入ㄌㄌ的懷抱。陷入懷抱之後，BB 就不會再丟骰子了。

現在，BB 想要算出，他期望要過多少天，才會陷入ㄌㄌ的懷抱。注意 BB 只要成功連續依序跟任何一組ㄌㄌ組合玩，他就會陷入ㄌㄌ的懷抱。

Input Format

輸入的第一行有一個正整數 N ，代表ㄌㄌ組合的個數。

接下來的 N 行，每行包含一個字串 s_i ，代表ㄌㄌ組合。

- $\sum |s_i| \leq 150$ 。

Output Format

請求出 BB 期望要過多少天，才會陷入ㄌㄌ的懷抱。已知這個天數可以表示成最簡分數 $\frac{P}{Q}$ ，請輸出 $P \cdot Q^{-1} \bmod 998244353$ 的值。

Sample Input

copy

Sample Input #1
1
121

Sample Input #2
1
6565

Sample Input #3
5
12
23
34
45
56

Sample Output

copy

Sample Output #1
222

Sample Output #2
1332

Sample Output #3
317138727

Hints

Problem Source


Subtasks		
No.	Testdata Range	Score

Testdata and Limits					⌵
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1000	262144	65536		
1	1000	262144	65536		
2	1000	262144	65536		
3	1000	262144	65536		
4	1000	262144	65536		
5	1000	262144	65536		
6	1000	262144	65536		
7	1000	262144	65536		
8	1000	262144	65536		
9	1000	262144	65536		
10	1000	262144	65536		
11	1000	262144	65536		
12	1000	262144	65536		
13	1000	262144	65536		
14	1000	262144	65536		
15	1000	262144	65536		
16	1000	262144	65536		
17	1000	262144	65536		
18	1000	262144	65536		
19	1000	262144	65536		
20	1000	262144	65536		
21	1000	262144	65536		
22	1000	262144	65536		
23	1000	262144	65536		
24	1000	262144	65536		
25	1000	262144	65536		
26	1000	262144	65536		
27	1000	262144	65536		
28	1000	262144	65536		
29	1000	262144	65536		
30	1000	262144	65536		
31	1000	262144	65536		
32	1000	262144	65536		

50. 鬼轉影片

SubmitStatusRanklistBack to Problems List

TopCoder



nevikw39
nevikw³⁹

User's AC Ratio80.0% (4/5)

Submission's AC Ratio80.0% (4/5)

Tags

Description

撰寫歌詞一直以來都是一個辛苦的工作，身為 Aqours 中負責寫歌詞的高海千歌，每次都要絞盡腦汁花費大量心思才能寫出一個符合期待的歌詞，尤其是在寫了這麼多歌曲後，想要能有突破性的創作實在是件困難的事。

最近的她為了寫歌，每天都吃不下飯，讓人非常心疼。更糟糕的是明天就是約定的交稿日了，千歌還是寫不出一個能夠溫暖人心卻又不失 Aqours 風格的歌詞，於是她向你求助，希望你能幫助她完成歌曲。

而你除了認識千歌外，也剛好與聖翔音樂學園的神樂光非常要好，因為平常神樂光也會向你討論歌詞的內容，所以你其實也有小光所寫的許多歌詞，於是你決定融合兩者的風格，把千歌給你的草稿中的一小部分歌詞換成小光寫的歌詞。

然而為了不要讓人一眼識破你其實是抄襲他人的歌曲，你在把小光的歌詞放入千歌的歌曲時，會希望小光的歌詞與千歌被換掉的那一部分歌詞的「最長共同前綴長度加上最長共同後綴長度」越長越好，這樣的話相信粉絲們如果不認真檢查應該都檢查不出來。此外，為了搭配樂曲本身，你也希望被換掉的歌詞的字數要與小光的歌詞字數一樣長。

現在你手上有著千歌給你的草稿了，而你也精心挑選的一小段小光的歌詞了，你可以把千歌歌詞中的任一段換成小光的歌詞，現在貴想知道千歌被換掉的歌詞與小光的歌詞的「最長共同前綴長度加上最長共同後綴長度」最長可以是多長。特別的，如果兩段歌詞完全一樣，此時這個長度被定義為原本小光的歌詞的長度。

請你告訴貴這個問題的答案吧！（你肯定不會讓貴失望的吧？）

Input Format

第一行輸入兩個整數 n, m 分別代表千歌的草稿的長度和小光的歌詞的長度。

第二行輸入一個字串，代表千歌的草稿。

第三行輸入一個字串，代表小光的歌詞。

- $1 \leq m \leq n \leq 10^6$
- 保證輸入的字串都由小寫英文字母構成。

Output Format

輸出一個整數，代表千歌被換掉的歌詞與小光的歌詞的「最長共同前綴長度加上最長共同後綴長度」最長可以是多長。

Sample Input

copy

Sample Input #1

```
5 3
aacbb
adb
```

Sample Input #2

```
5 3
abbba
bbb
```

Sample Output

copy

Sample Output #1

```
2
```

Sample Output #2

```
3
```

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⤴
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	
7	1000	262144	65536	
8	1000	262144	65536	
9	1000	262144	65536	
10	1000	262144	65536	
11	1000	262144	65536	
12	1000	262144	65536	
13	1000	262144	65536	
14	1000	262144	65536	
15	1000	262144	65536	
16	1000	262144	65536	
17	1000	262144	65536	
18	1000	262144	65536	

✓46. IOIC 王國

SubmitStatusRanklistBack to Problems List

TopCoder



dumbing

數數

User's AC Ratio

100.0% (3/3)

Submission's AC Ratio

42.9% (3/7)

Tags

Description

IOIC 王國是一個獨裁的國家，由帥哥總召 Robert 統治，這個國家一共有 N 座城市以及 M 條道路，每一條道路連接著兩座不同的城市。帥哥總召 Robert 為了監控整個王國，研發了 K 種不同類型的機器人 (每個種類都有無數多個機器人)。他想要將機器人分配到每一條道路上，讓每一條道路都恰有一個機器人監控。

為了增加管理王國的趣味性，總召會進行一些操作，操作的方式如下：總召每次可以選擇一些不同的城市，編號分別為 x_1, x_2, \dots, x_t ，使得對所有 $1 \leq i \leq t$ ，編號 x_i 與 x_{i+1} 的城市之間都有道路連接，其中 $x_{t+1} = x_1$ 。接著，總召會旋轉這些道路上的機器人，也就是說在編號 x_i 與 x_{i+1} 的城市之間的機器人，會移動到連接著編號 x_{i+1} 與 x_{i+2} 的城市之間的道路。

總召很好奇，在這樣的情況下，最多可以選出幾種不同的分配方式，使得對於任意兩種分配方式，都沒有辦法經過若干次操作從一種分配方式變成另一種分配方式？如果存在一條道路在兩種分配方式中被賦予不同種類的機器人，我們就說這兩種分配方式是不同的。

Input Format

輸入的第一行包含三個正整數 N, M, K ，分別代表城市的數量，道路的數量，以及機器人類型的數量。接下來的 M 行，每一行都有兩個正整數 x, y ($1 \leq x, y \leq N, x \neq y$)，代表這條道路連接著第 x 座城市與第 y 座城市，保證任意兩座城市之間不會有兩條以上的道路連接。

- $1 \leq N, M \leq 5 \times 10^5$
- $1 \leq K \leq 10^9$

Output Format

請輸出一個整數代表分配的方案數，答案請模 998244353 之後輸出。

Sample Input

Sample Input #1

4 4 2
1 2
2 3
3 1
2 4

Sample Input #2

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

Sample Input #3

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

Sample Input #4

10 8 4
9 4
5 3
6 4
2 4
6 10
4 7
9 2
4 10

Sample Output

Sample Output #1

8

Sample Output #2

121

Sample Output #3

6

Sample Output #4

9216

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	2000	262144	65536	
1	2000	262144	65536	
2	2000	262144	65536	
3	2000	262144	65536	
4	2000	262144	65536	
5	2000	262144	65536	
6	2000	262144	65536	
7	2000	262144	65536	
8	2000	262144	65536	
9	2000	262144	65536	
10	2000	262144	65536	
11	2000	262144	65536	
12	2000	262144	65536	
13	2000	262144	65536	
14	2000	262144	65536	
15	2000	262144	65536	
16	2000	262144	65536	
17	2000	262144	65536	
18	2000	262144	65536	
19	2000	262144	65536	
20	2000	262144	65536	
21	2000	262144	65536	
22	2000	262144	65536	
23	2000	262144	65536	
24	2000	262144	65536	
25	2000	262144	65536	
26	2000	262144	65536	
27	2000	262144	65536	
28	2000	262144	65536	
29	2000	262144	65536	
30	2000	262144	65536	
31	2000	262144	65536	
32	2000	262144	65536	
33	2000	262144	65536	
34	2000	262144	65536	
35	2000	262144	65536	
36	2000	262144	65536	
37	2000	262144	65536	
38	2000	262144	65536	
39	2000	262144	65536	
40	2000	262144	65536	
41	2000	262144	65536	
42	2000	262144	65536	
43	2000	262144	65536	
44	2000	262144	65536	
45	2000	262144	65536	
46	2000	262144	65536	
47	2000	262144	65536	
48	2000	262144	65536	
49	2000	262144	65536	
50	2000	262144	65536	
51	2000	262144	65536	
52	2000	262144	65536	
53	2000	262144	65536	
54	2000	262144	65536	
55	2000	262144	65536	
56	2000	262144	65536	
57	2000	262144	65536	
58	2000	262144	65536	
59	2000	262144	65536	
60	2000	262144	65536	
61	2000	262144	65536	
62	2000	262144	65536	
63	2000	262144	65536	
64	2000	262144	65536	
65	2000	262144	65536	
66	2000	262144	65536	
67	2000	262144	65536	
68	2000	262144	65536	
69	2000	262144	65536	
70	2000	262144	65536	
71	2000	262144	65536	
72	2000	262144	65536	
73	2000	262144	65536	
74	2000	262144	65536	
75	2000	262144	65536	
76	2000	262144	65536	
77	2000	262144	65536	
78	2000	262144	65536	
79	2000	262144	65536	
80	2000	262144	65536	
81	2000	262144	65536	
82	2000	262144	65536	
83	2000	262144	65536	
84	2000	262144	65536	
85	2000	262144	65536	
86	2000	262144	65536	
87	2000	262144	65536	
88	2000	262144	65536	
89	2000	262144	65536	
90	2000	262144	65536	
91	2000	262144	65536	
92	2000	262144	65536	
93	2000	262144	65536	
94	2000	262144	65536	
95	2000	262144	65536	
96	2000	262144	65536	
97	2000	262144	65536	
98	2000	262144	65536	

49. 時代的卷跡

Submit

Status

Ranklist

Back to Problems List

TopCoder

User's AC Ratio
NaN% (0/0)

Submission's AC Ratio
NaN% (0/0)

Tags

Description

貴育成了 $3N + 1$ 位賽馬娘，編號由 0 到 $3N$ 。

賽馬娘們有三種主要的營養來源：白飯、漢堡排、紅蘿蔔。

編號為 i 的賽馬娘在每餐都必定會食用總共 i 份的白米飯、漢堡排、紅蘿蔔，而且每種餐點都一定會食用整數份。

編號為 i 的賽馬娘在時間 t 時（這裡的時間是相對時間，所以 t 可能為任意實數）的固有馬力為 it 。

若在時間 t 舉辦一場賽跑，每位賽馬娘都會在時間 t 前預先吃飽（即，貴會讓編號 i 的賽馬娘任意食用白米飯、漢堡排、紅蘿蔔再讓她在時間 t 參賽）。

一位賽馬娘食用了 i 份白米飯就會在她當前的固有馬力之上額外獲得 f_i 大小的馬力。

一位賽馬娘食用了 i 份漢堡排就會在她當前的固有馬力之上額外獲得 g_i 大小的馬力。

一位賽馬娘食用了 i 份紅蘿蔔就會在她當前的固有馬力之上額外獲得 h_i 大小的馬力。

也就是說，編號為 i 的賽馬娘在於時間 t 舉辦的賽跑中所能表現出的馬力是 $it + \max_{0 \leq a, b, c \leq N, a+b+c=i} (f_a + g_b + h_c)$ 。

在時間 t 舉辦的賽跑的勝者，就是在時間 t 時馬力最大的賽馬娘，若有不只一名賽馬娘達到最大的馬力，就會判定為沒有勝者。

貴想去特別關心那些從未取勝的賽馬娘，也就是說貴想知道那些無論在任何時間 t 賽跑都無法取勝的賽馬娘們她們的編號。

請你告訴貴這個問題的答案吧！（你肯定不會讓貴失望的吧？）

Input Format

第一行輸入一個正整數 $N \leq 3 \times 10^5$ ，代表貴育成了 $3N + 1$ 位賽馬娘。

第二行輸入 $N + 1$ 個整數，分別代表 f_0 到 f_N 。

第三行輸入 $N + 1$ 個整數，分別代表 g_0 到 g_N 。

第四行輸入 $N + 1$ 個整數，分別代表 h_0 到 h_N 。

$0 \leq f_i, g_i, h_i \leq 10^9$

Output Format

第一行輸出一個整數 M ，代表從未取勝的賽馬娘的個數。

第二行輸出 M 個整數，代表所有從未取勝的賽馬娘的編號從小到大排序後的結果。

Sample Input

copy

Sample Input #1

```
3
3 1 8 7
9 1 3 1
5 1 1 6
```

Sample Input #2

```
1
1 2
1 2
1 2
```

Sample Output

copy

Sample Output #1

```
5
1 3 4 7 8
```

Sample Output #2

```
2
1 2
```

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⤴
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	
7	1000	262144	65536	
8	1000	262144	65536	
9	1000	262144	65536	
10	1000	262144	65536	
11	1000	262144	65536	
12	1000	262144	65536	
13	1000	262144	65536	
14	1000	262144	65536	
15	1000	262144	65536	
16	1000	262144	65536	
17	1000	262144	65536	
18	1000	262144	65536	
19	1000	262144	65536	
20	1000	262144	65536	
21	1000	262144	65536	
22	1000	262144	65536	
23	1000	262144	65536	
24	1000	262144	65536	
25	1000	262144	65536	
26	1000	262144	65536	
27	1000	262144	65536	
28	1000	262144	65536	
29	1000	262144	65536	
30	1000	262144	65536	
31	1000	262144	65536	
32	1000	262144	65536	
33	1000	262144	65536	
34	1000	262144	65536	
35	1000	262144	65536	
36	1000	262144	65536	
37	1000	262144	65536	
38	1000	262144	65536	
39	1000	262144	65536	
40	1000	262144	65536	
41	1000	262144	65536	

44. 黑洞危機

[Submit](#)[Status](#)[Ranklist](#)[Back to Problems List](#)

TopCoder

User's AC Ratio

NaN% (0/0)

Submission's AC Ratio

NaN% (0/0)

Tags

Description

西元 2147483647 年，若干變形黑洞開始在世界上到處出現，這些黑洞會將所有人們往它身上吸引過去，造成了前所未有的危機！

你，身為一個資訊科學家，其實根本不在意人們的死活，你只在乎排在 $x = 1 \sim N$ 上， N 個人的行為，他們由於太過於恐懼，無法離開自己所在的 x 座標，只能上下移動來改變他們的 y 座標，而他們的 y 座標一開始是 $y_1, y_2 \cdots, y_N$ 。

你希望能夠寫一支程式，用來記錄並模擬依序發生的事件，好讓你能夠在途中隨時得到你想要的資訊。

事件的種類如下：

- $1\ l\ r\ p$ ：一個變形黑洞出現了！他形如一條出現在 (l, p) 到 (r, p) 的線段，並將所有 x 座標介在 $[l, r]$ 的人的 y 座標 y_i ，改成 $f_p(y_i)$ 。更改完畢後，這個黑洞就會消失。
- $2\ l\ r\ d$ ： x 座標介在 $[l, r]$ 的人產生了恐懼，他們決定一起平移，也就是讓自己的 y 座標加上 d (d 可能為負)。
- $3\ l\ r$ ：你很好奇 x 座標介在 $[l, r]$ 的人的 y 座標總和，請輸出之。
- $4\ l\ r$ ：你很好奇 x 座標介在 $[l, r]$ 的人的 y 座標最大值，請輸出之。
- $5\ l\ r$ ：你很好奇 x 座標介在 $[l, r]$ 的人的 y 座標最小值，請輸出之。

其中， $f_p(y) = \begin{cases} p + \lfloor \sqrt{y-p} \rfloor & \text{if } y \geq p \\ p - \lfloor \sqrt{p-y} \rfloor & \text{if } y < p \end{cases}$ ，意即變形黑洞會嘗試把指定區間的人全部往自己吸引過去。

Input Format

輸出首行有兩個正整數 N, Q ，代表你觀察的人數以及事件的數量。

接下來一行共 N 個數字 $y_1, y_2 \cdots, y_N$ ，代表 x 座標是 i 的人， y 座標一開始是 y_i 。

接下來 Q 行依序是事件發生的順序，每行先是一個 k 代表事件種類，接著兩個正整數 l, r ，代表事件所對應到的區間，最後可能還有一個整數 p 或 d ，代表第一種或第二種事件的額外參數。

所有同行相鄰數字以單一空格隔開。

- $1 \leq N, Q \leq 2 \times 10^5$
- $0 \leq |y_i| \leq 10^7$
- $1 \leq k \leq 5$
- $1 \leq l \leq r \leq N$
- $0 \leq |p|, |d| \leq 10^7$

Output Format

對於每筆你好奇的資訊，請輸出正確的答案於一行。

Sample Input

copy

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

Sample Output

copy

```
16
2
13
3
-9
```

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	2000	262144	65536	
1	2000	262144	65536	
2	2000	262144	65536	
3	2000	262144	65536	
4	2000	262144	65536	
5	2000	262144	65536	
6	2000	262144	65536	
7	2000	262144	65536	
8	2000	262144	65536	
9	2000	262144	65536	
10	2000	262144	65536	
11	2000	262144	65536	
12	2000	262144	65536	
13	2000	262144	65536	
14	2000	262144	65536	
15	2000	262144	65536	
16	2000	262144	65536	
17	2000	262144	65536	
18	2000	262144	65536	
19	2000	262144	65536	
20	2000	262144	65536	
21	2000	262144	65536	
22	2000	262144	65536	
23	2000	262144	65536	
24	2000	262144	65536	
25	2000	262144	65536	
26	2000	262144	65536	
27	2000	262144	65536	
28	2000	262144	65536	
29	2000	262144	65536	
30	2000	262144	65536	
31	2000	262144	65536	
32	2000	262144	65536	
33	2000	262144	65536	
34	2000	262144	65536	
35	2000	262144	65536	