

Universe cup 3 stage 33 : India



Dịch đề bài:

Bài A: AB thành C

Giới hạn thời gian: 1 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Bạn được cho một chuỗi chỉ gồm các ký tự **A**, **B** hoặc **C**. Bạn có thể thực hiện phép biến đổi sau **bao nhiêu lần tùy ý**:

- Chọn hai vị trí $1 \leq i < j \leq |S|$ sao cho $S_i \neq S_j$.
- Xóa S_i và S_j .
- Thêm vào chuỗi một ký tự **bổ sung** của cặp (S_i, S_j) tại **bất kỳ vị trí nào**.

Ký tự bổ sung được định nghĩa như sau:

- Nếu (A, B) hoặc $(B, A) \rightarrow$ ký tự bổ sung là **C**.
- Nếu (B, C) hoặc $(C, B) \rightarrow$ ký tự bổ sung là **A**.
- Nếu (A, C) hoặc $(C, A) \rightarrow$ ký tự bổ sung là **B**.

Bạn được phép thực hiện bao nhiêu phép biến đổi tùy thích. Nhiệm vụ của bạn là **tìm ra chuỗi sau cùng nhỏ nhất theo thứ tự từ điển**.

Dữ liệu vào:

- Dòng đầu tiên là số lượng bộ test **T**.

- Mỗi test gồm:
 - Một số nguyên **N**: độ dài chuỗi.
 - Một chuỗi **S** gồm **N** ký tự **A**, **B**, hoặc **C**.

Dữ liệu ra:

- Với mỗi test, in ra chuỗi nhỏ nhất theo thứ tự từ điển có thể đạt được sau khi biến đổi.
-

Ràng buộc:

- $1 \leq T \leq 10^5$
 - $1 \leq N \leq 2 \cdot 10^5$
 - Tổng độ dài chuỗi qua tất cả test không vượt quá 10^6 .
-

Ví dụ:

Input :

3
2
BC
3
AAA
3
CAA

Output :

A

AAA

AB

Giải thích:

- Test 1: **BC** → xóa **B** và **C**, chèn **A** → kết quả là **A**.
- Test 2: **AAA** → không có cặp ký tự khác nhau nào để biến đổi → giữ nguyên **AAA**.
- Test 3: **CAA** → không thể biến đổi thêm → giữ nguyên **AB** (hoặc **BA**, nhưng **AB** nhỏ hơn từ điển).

Tóm tắt:

- Cho chuỗi gồm **A, B, C**.
- Lặp lại thao tác: chọn hai ký tự **khác nhau**, xóa chúng, thêm ký tự **bổ sung**.
- Yêu cầu: tìm chuỗi **nhỏ nhất từ điển** sau cùng có thể đạt được.



Dịch đề bài:

Bài B: Xếp vật vào thùng

Giới hạn thời gian: 5 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Hôm nay chúng ta sẽ giải một bài toán **NP-Hard**.

Bạn có **N** vật thể, vật thứ **i** có kích thước **A_i** ($1 \leq A_i \leq 12$).

Bạn có **nhiều thùng**, mỗi thùng có sức chứa tối đa **12**.



Nhiều vật thể có thể được đặt cùng một thùng **nếu tổng kích thước của chúng không vượt quá 12**.

Nhiệm vụ của bạn là tìm **số lượng thùng ít nhất** cần dùng để chứa tất cả các vật.

Ràng buộc đặc biệt:

- Với **các test chính (không phải test mẫu)**:
 - Số lượng test **$T = 100$** .
 - Số lượng vật thể trong mỗi test **$N = 1000$** .
 - Kích thước mỗi vật **A_i** được sinh ngẫu nhiên theo phân phối đều từ **1 đến 12**.

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T** : số lượng test.
- Mỗi test gồm:
 - Một số nguyên **N** : số vật thể.
 - Một dòng gồm **N** số nguyên **A_1, A_2, \dots, A_N** : kích thước từng vật.

Dữ liệu ra:

- Với mỗi test, in ra số lượng thùng ít nhất cần dùng.

Ràng buộc:

- Với các test chính:
 - $T = 100, N = 1000$** .
 - $1 \leq A_i \leq 12$** .
- Có đúng **19 test chính (không phải mẫu)**.

Ví dụ:

Input:

```
10
5
8 2 2 3 9
3
1 1 1
8
9 8 2 3 4 2 2 1 2
10
2 2 2 2 2 2 3 3 3 3
6
5 2 8 4 1 1
8
12 1 2 9 8 2 8 4 4
5
5 5 6 6 6
8
10 1 2 5 4 8 2 2 1
7
5 1 5 7 9 4 4
6
12 1 1 9 3 5
```

Output:

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

Giải thích Test 1:

- Có thể phân phối như sau:
 - Thùng 1: Vật 1, 2, 3 (tổng = 12).
 - Thùng 2: Vật 4, 5 (tổng = 12).
 - Cần **2 thùng**.
-

Tóm tắt:

- Cho **N vật**, mỗi vật có kích thước từ **1 đến 12**.
- Mỗi thùng chứa tối đa **12**.
- Nhiệm vụ: tìm **số thùng ít nhất** để chứa hết các vật (tổng kích thước mỗi thùng ≤ 12).



Dịch đề bài:

Bài C: Tạo chuỗi uwu

Giới hạn thời gian: 4 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Bạn được cho một số nguyên dương **N**.

Hãy **tạo ra một chuỗi ngắn nhất** chỉ gồm các ký tự '**u**' và '**w**', sao cho thỏa mãn:

- Số lượng **dây con** (không nhất thiết liên tiếp) của chuỗi bằng "**uwu**" **chính xác bằng N**.

Lưu ý: Dãy con là một tập hợp các ký tự được lấy từ chuỗi ban đầu, giữ nguyên thứ tự nhưng không cần liên tiếp.

Bài toán đảm bảo rằng **luôn tồn tại ít nhất một chuỗi thỏa mãn yêu cầu**. Nhiệm vụ của bạn là tìm **chuỗi ngắn nhất** có thể.

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T**: số lượng test.
- Mỗi test gồm:
 - Một số nguyên **N**: số dãy con "**uwu**" cần có.

Dữ liệu ra:

- Với mỗi test, in ra một chuỗi ngắn nhất thỏa mãn số dãy con "**uwu**" đúng bằng **N**.
-

Ràng buộc:

- $1 \leq T \leq 10^3$
 - $1 \leq N \leq 10^{18}$
 - Tổng độ dài các chuỗi ngắn nhất qua tất cả test không vượt quá 10^7 .
-

Ví dụ:

Input :

6

1
2
3
4
5
6

Output:

uwu
uwwu
uwwwu
uwwuu
uwwwwwu
uwwwu

Giải thích Test 1:

- Chuỗi "**uwu**" có đúng **1** dãy con "**uwu**" là chính nó.

Giải thích Test 2:

- Chuỗi "**uwwu**" có **2** dãy con "**uwu**":
 - Lấy ký tự ở vị trí (1,2,4).
 - Lấy ký tự ở vị trí (1,3,4).

Tóm tắt:

- Cho số nguyên **N**.
- Tạo chuỗi ngắn nhất chỉ gồm '**u**' và '**w**', sao cho có đúng **N dãy con "uwu"** (không cần liên tiếp).

- Mục tiêu: **rút gọn độ dài chuỗi** nhưng vẫn đảm bảo đúng **N** dãy con "**uwu**".



Dịch đề bài:

Bài D: Đếm các mảng khoảng cách

Giới hạn thời gian: 2 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Cho một **cây** có **N đỉnh**, được đánh số từ **1** đến **N**, và **một tập con các đỉnh S**.

Ta định nghĩa **mảng A** gồm **N phần tử** như sau:

- Với mỗi đỉnh **i**,
 $A_i = \max(\text{dist}(x, i))$ với **$x \in S$** .
(Trong đó, **$\text{dist}(x, i)$** là số cạnh trên đường đi ngắn nhất duy nhất từ **x** đến **i**).

Có tất cả **$2^N - 1$** tập con **S** khác rỗng có thể chọn.

Nhiệm vụ của bạn là **tìm số lượng mảng A khác nhau** có thể tạo ra khi xét tất cả các tập con **S**.

Vì kết quả có thể rất lớn, hãy in ra kết quả **modulo 998244353**.

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T**: số lượng test.
- Mỗi test gồm:
 - Một số nguyên **N**: số đỉnh của cây.
 - **N-1 dòng**, mỗi dòng gồm 2 số nguyên **u, v**: biểu diễn một cạnh nối **u** và **v**.

Dữ liệu ra:

- Với mỗi test, in ra số lượng mảng **A** khác nhau có thể tạo ra (modulo **998244353**).

Ràng buộc:

- $1 \leq T \leq 10^4$.
- $2 \leq N \leq 2 \cdot 10^5$.
- Tổng tất cả các **N** qua các test không vượt quá $2 \cdot 10^5$.
- Các cạnh tạo thành **một cây**.

Ví dụ:

Input :

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

Output :

```
3
```

6
23

Giải thích Test 1:

- Có 3 tập con **S** có thể chọn:
 - **S = {1}** → A = [0, 1]
 - **S = {2}** → A = [1, 0]
 - **S = {1, 2}** → A = [1, 1]
- Tổng cộng có **3 mảng A khác nhau**.

Tóm tắt:

- Cho **cây N đỉnh**.
- Với mọi tập con **S** (khác rỗng), tạo ra mảng **A**:
 - **A_i** là khoảng cách xa nhất từ **i** đến các đỉnh trong **S**.
- Đếm số lượng **mảng A khác nhau** có thể tạo ra khi xét tất cả tập **S**.
- In ra **kết quả modulo 998244353**.



Dịch đề bài:

Bài E: Bài toán đếm dễ

Giới hạn thời gian: 3 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Cho một **mảng A**. Bạn có thể thực hiện **phép biến đổi** sau không giới hạn số lần:

- Chọn một chỉ số **i** ($1 \leq i \leq |A|$) sao cho **A[i] = i**, sau đó **xóa phần tử thứ i** khỏi mảng.
- Cụ thể, sau khi xóa phần tử thứ **i**, mảng mới sẽ là:
A = A[1, i-1] + A[i+1, |A|]
(Trong đó, **A[L, R]** là đoạn con từ **L** đến **R**, và **+** là phép nối mảng).

Gọi **f(A)** là **độ dài nhỏ nhất** của mảng **A** sau khi thực hiện các phép biến đổi **tối ưu**.

Bạn được cho hai số nguyên **N** và **M**.

Hãy tính tổng **f(A)** của **tất cả các mảng A** có:

- Độ dài N.**
- $1 \leq A[i] \leq M$ với mọi $1 \leq i \leq N$.

Vì kết quả có thể rất lớn, hãy in ra **kết quả modulo 998244353**.

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T**: số lượng test.
- Mỗi test gồm:
 - Hai số nguyên **N** và **M**.

Dữ liệu ra:

- Với mỗi test, in ra tổng **f(A)** của tất cả các mảng **A** thỏa mãn điều kiện (modulo **998244353**).

Ràng buộc:

- $1 \leq T \leq 10$.
 - $1 \leq N, M < 998244353$.
 - $|N - M| \leq 2000$.
-

Ví dụ:

Input :

```
9
2 2
2 1
3 1
3 2
3 3
3 5
6 3
23 10
900000000 900000000
```

Output :

```
3
0
0
9
44
284
2534
243195864
193285282
```

Giải thích Test 1:

- Với $N = 2, M = 2 \rightarrow$ có tất cả $2^2 = 4$ mảng:

1. **[1, 1]** → xóa vị trí 1 → còn **[1]**, tiếp tục xóa → còn **[]** → **f = 0**.

2. **[1, 2]** → xóa vị trí 2 → còn **[1]**, tiếp tục xóa → còn **[]** → **f = 0**.

3. **[2, 1]** → không có vị trí nào thỏa mãn → **f = 2**.

4. **[2, 2]** → xóa vị trí 2 → còn **[2]** → không thể xóa tiếp → **f = 1**.

- Tổng: **0 + 0 + 2 + 1 = 3**.

Tóm tắt:

- Cho mảng **A** độ dài **N**, các phần tử **$1 \leq A[i] \leq M$** .
- Phép biến đổi: xóa phần tử ở vị trí **i** nếu **$A[i] = i$** .
- Tính tổng **f(A)** cho tất cả các mảng có thể tạo ra.
- In kết quả **modulo 998244353**.



Dịch đề bài:

Bài F: Giá cả tham lam

Giới hạn thời gian: 2 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Có **N món đồ** được đem bán đấu giá. Tuy nhiên, cuộc đấu giá này rất **tham lam** và **giá của từng món đồ thay đổi tùy thuộc vào số tiền bạn đã chi tiêu trước đó**.
Cụ thể:

- Món đồ thứ **i** có hai tham số:
 - **Mi** (hệ số tuyến tính).

- **C_i** (hằng số).
- **Giá của món đồ thứ i** được tính bằng công thức:
M_i · X + C_i,
trong đó **X** là **tổng số tiền bạn đã chi trước đó**.
- Mỗi món đồ **chỉ mua được một lần**.
- Bạn **được phép quyết định thứ tự mua các món đồ**.

Bạn cần trả lời **Q truy vấn**, mỗi truy vấn hỏi:

- Với ngân sách **P_i**, bạn có thể mua **tối đa bao nhiêu món đồ?**
-

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T**: số lượng test.
 - Mỗi test gồm:
 - Hai số nguyên **N** và **Q**: số món đồ và số truy vấn.
 - Dòng thứ hai gồm **N** số nguyên: **M₁, M₂, ..., M_N** (hệ số tuyến tính của từng món đồ).
 - Dòng thứ ba gồm **N** số nguyên: **C₁, C₂, ..., C_N** (hằng số của từng món đồ).
 - **Q dòng tiếp theo**, mỗi dòng một số nguyên **P_i**: ngân sách cho mỗi truy vấn.
-

Dữ liệu ra:

- Với mỗi test, in ra **Q số nguyên**: câu trả lời cho từng truy vấn (tối đa số món đồ mua được với từng ngân sách).

Ràng buộc:

- $1 \leq T \leq 10^4$.
 - $1 \leq N, Q \leq 2 \cdot 10^5$.
 - $0 \leq M_i, C_i, P_i \leq 10^9$.
 - Tổng tất cả **N** và **Q** trên các test không vượt quá $2 \cdot 10^5$.
-

Ví dụ:

Input :

```
2
3 4
1 1 0
3 6 4
2
6
7
19
6 6
0 0 1 1 2 3
0 1 0 2 3 4
0
1
8
15
3
10000
```

Output :

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


Giải thích Test 1:

- **Ngân sách 2:** Không thể mua món nào.
- **Ngân sách 6:**
 - Có thể mua 1 món bất kỳ:
 - Món 1: giá **3** (vì $X=0$, $1 \cdot 0 + 3$).
 - Món 2: giá **6** ($1 \cdot 0 + 6$).
 - Món 3: giá **4** ($0 \cdot 0 + 4$).
 - Nhưng chỉ đủ mua 1 món.
- **Ngân sách 7:**
 - Mua món 1 trước (giá **3**), còn **4**.
 - Mua món 3 (giá **4**), hết ngân sách.
 - Tổng cộng mua được **2 món**.
- **Ngân sách 19:**
 - Mua món 2 trước (giá **6**), còn **13**.
 - Mua món 1 (giá **9**), còn **4**.
 - Mua món 3 (giá **4**), hết ngân sách.
 - Mua được **3 món**.

Tóm tắt:

- Có **N món đồ**, mỗi món có giá **$M_i \cdot X + C_i$** , với **X** là tổng tiền đã chi trước đó.
- **Q truy vấn**, mỗi truy vấn hỏi: Với ngân sách **P_i**, mua tối đa được bao nhiêu món.
- Mỗi món **chỉ mua 1 lần**, quyết định thứ tự mua tùy ý.



Dịch đề bài:

Bài G: Xổ số từ điển

Giới hạn thời gian: 2 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Bạn có một chuỗi **S** độ dài **N** gồm các ký tự tiếng Anh thường.

Quá trình **raffle(L, R)** (với $1 \leq L < R \leq N$) được định nghĩa như sau:

- **Bước 1:**
 - **X** là đoạn con **S[L, R-1]** (từ **S_L** đến **S_{R-1}**).
 - **Y** là đoạn con **S[L+1, R]** (từ **S_{L+1}** đến **S_R**).
- **Bước 2:**
 - Nếu **Y < X** (so sánh từ điển), **tăng L lên 1**.
 - Ngược lại, **giảm R đi 1**.
- **Bước 3:**
 - Nếu **L = R**, dừng quá trình và **trả về L**.
 - Ngược lại, quay lại **Bước 1**.

Bạn được cho **Q truy vấn**, mỗi truy vấn gồm hai số **L** và **R**.

Với mỗi truy vấn, hãy tìm **giá trị cuối cùng của L** sau khi thực hiện quá trình **raffle(L, R)**.

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T**: số lượng test.
 - Mỗi test gồm:
 - Hai số nguyên **N** và **Q**: độ dài chuỗi và số truy vấn.
 - Một chuỗi **S** độ dài **N**.
 - **Q dòng**, mỗi dòng hai số nguyên **L** và **R**: mô tả một truy vấn.
-

Dữ liệu ra:

- Với mỗi truy vấn, in ra **giá trị cuối cùng của L**.
-

Ràng buộc:

- $1 \leq T \leq 10^4$.
 - $2 \leq N \leq 2 \cdot 10^5$.
 - $1 \leq Q \leq 2 \cdot 10^5$.
 - Tổng tất cả **N** và tổng tất cả **Q** trên các test không vượt quá $2 \cdot 10^5$.
 - **S** chỉ chứa các chữ cái thường ('a'-'z').
-

Ví dụ:

Input:

```
4
6 3
kanpur
1 6
4 6
5 6
10 2
adccbabbab
1 10
2 9
5 3
accaa
1 5
2 4
3 5
5 1
jddda
3 4
```

Output:

```
2
4
6
1
6
1
4
4
3
```

Giải thích test 1, truy vấn 1 (L=1, R=6):

- **X = "kanpu", Y = "anpur" → Y < X → tăng L thành 2.**
- **X = "anpu", Y = "npur" → Y > X → giảm R thành 5.**
- **X = "anp", Y = "npu" → Y > X → giảm R thành 4.**

- $X = \text{"an"}, Y = \text{"np"} \rightarrow Y > X \rightarrow$ giảm R thành 3 .
 - $X = \text{"a"}, Y = \text{"n"} \rightarrow Y > X \rightarrow$ giảm R thành 2 .
 - Kết thúc: $L = R = 2$.
-

Tóm tắt:

- Cho chuỗi S độ dài N .
- Với mỗi truy vấn (L, R) , lặp lại:
 - So sánh đoạn con $X = S[L, R-1]$ và $Y = S[L+1, R]$.
 - Nếu $Y < X$, $L++$; ngược lại $R--$.
- Dừng khi $L = R$, trả về L .



Dịch đề bài:

Bài H: Đồ thị đa số

Giới hạn thời gian: 4 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Bạn được cho một **mảng A** gồm **N phần tử**: A_1, A_2, \dots, A_N .

Xây dựng một **đồ thị G** với **N đỉnh**, mỗi đỉnh tương ứng với một vị trí trong mảng.

Vẽ cạnh (i, j) nếu **thỏa mãn hai điều kiện** sau:

- $1 \leq i < j \leq N$.
- Đoạn con $A[i, j]$ có **phần tử chiếm đa số** (majority element).

Định nghĩa:

- Đoạn con **$A[i, j]$** là mảng **$[A_i, A_{i+1}, \dots, A_j]$** .
- Một mảng có **phần tử chiếm đa số** nếu tồn tại một phần tử xuất hiện **nhiều hơn $M/2$ lần**, với **M** là độ dài mảng.

Nhiệm vụ:

Tìm **số lượng thành phần liên thông** của đồ thị **G** .

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T** : số lượng test.
- Mỗi test gồm:
 - Một số nguyên **N** : số phần tử của mảng (và số đỉnh của đồ thị).
 - Một dòng gồm **N số nguyên: A_1, A_2, \dots, A_N** .

Dữ liệu ra:

- Với mỗi test, in ra **số lượng thành phần liên thông** của đồ thị **G** .
-

Ràng buộc:

- **$1 \leq T \leq 10^5$** .
- **$2 \leq N \leq 2 \cdot 10^6$** .
- **$1 \leq A_i \leq N$** .
- Tổng tất cả **N** qua các test không vượt quá **$2 \cdot 10^6$** .

Ví dụ:

Input:

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

Output:

```
2
4
1
1
```

Giải thích Test 1:

- Có **2 cạnh**:
 - **(1, 3)**: đoạn **$A[1, 3] = [1, 2, 1]$** , phần tử chiếm đa số là **1**.
 - **(2, 4)**: đoạn **$A[2, 4] = [2, 1, 2]$** , phần tử chiếm đa số là **2**.
- Đồ thị có **2 thành phần liên thông**.

Tóm tắt:

- Cho **mảng A** với **N phần tử**.
- Xây **đồ thị G**:
 - **Có cạnh (i, j)** nếu **đoạn A[i, j]** có phần tử chiếm đa số.
- Tính **số thành phần liên thông** trong **G**.



Dịch đề bài:

Bài 1: Cử tri chiếm đa số

Giới hạn thời gian: 1 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Có **N cử tri** xếp thành một hàng, được đánh số từ **1** đến **N**.

Mỗi người sẽ bỏ phiếu cho một trong hai ứng cử viên: **Alice** hoặc **Bob**.

- Ý kiến của cử tri được biểu diễn bằng chuỗi **S**:
 - **S[i] = A** nếu cử tri thứ **i** ủng hộ **Alice**.
 - **S[i] = B** nếu cử tri thứ **i** ủng hộ **Bob**.

Bạn muốn **giành xếp kết quả bầu cử để Alice thắng**.

Bạn có một năng lực đặc biệt:

- Bạn có thể **thuyết phục một số cử tri** thành **cử tri chiếm đa số**.

Quy tắc cử tri chiếm đa số:

- Một cử tri chiếm đa số tại vị trí **X** sẽ bỏ phiếu cho ứng viên đã có **nhiều phiếu hơn** tính đến vị trí **X-1**.
- Nếu số phiếu của hai ứng viên bằng nhau tại vị trí đó, cử tri chiếm đa số **sẽ không bỏ phiếu**.

Alice thắng nếu số phiếu cho cô ấy **lớn hơn hẳn** số phiếu cho Bob.

Gọi **$f(S)$** là **số lượng tối thiểu cử tri cần chuyển thành cử tri chiếm đa số để Alice thắng**.

- Nếu **không thể** làm Alice thắng **dù thay đổi cỡ nào**, thì **$f(S) = -1$** .
-

Nhiệm vụ:

- Cho chuỗi **S** và **Q truy vấn** dạng:
 - Với mỗi đoạn con **$S[L, R]$** , hãy tính **$f(S[L, R])$** .
-

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T**: số lượng test.
 - Mỗi test gồm:
 - Hai số nguyên **N** và **Q**: độ dài chuỗi **S** và số lượng truy vấn.
 - Một chuỗi **S** gồm các ký tự **A** hoặc **B**.
 - Q dòng**, mỗi dòng hai số nguyên **L** và **R**: mô tả đoạn con **$S[L, R]$** .
-

Dữ liệu ra:

- Với mỗi truy vấn, in ra giá trị **$f(S[L, R])$** .
-



Ràng buộc:

- $1 \leq T \leq 10^4$.
 - $1 \leq N, Q \leq 2 \cdot 10^5$.
 - $1 \leq L \leq R \leq N$.
 - Tổng tất cả **N** và **Q** trên các test **không vượt quá $2 \cdot 10^5$** .
-

Ví dụ:

Input :

```
2
4 4
AABA
1 4
2 3
3 3
3 4
5 2
BBBBA
1 5
5 5
```

Output :

```
0
1
-1
1
4
0
```

Giải thích Test 1:

- **Query 1 (1,4):**

- Alice có 3 phiếu, Bob có 1 phiếu → **Alice đã thắng**, nên **f = 0**.

- **Query 2 (2,3):**

- Alice 1 phiếu, Bob 1 phiếu → Nếu chuyển cử tri Bob thành **cử tri chiếm đa số**, thì:
 - Cử tri thứ 2 (Alice) giữ nguyên.
 - Cử tri thứ 3 (cử tri chiếm đa số, số phiếu trước đó là Alice:1, Bob:0 → bỏ cho Alice).
→ Alice 2, Bob 0 → **f = 1**.

- **Query 3 (3,3):**

- Một phiếu Bob duy nhất, không thể chuyển → **f = -1**.

- **Query 4 (3,4):**

- Alice 1 phiếu, Bob 1 phiếu → Nếu chuyển Bob thành **cử tri chiếm đa số**:
 - Bob sẽ không bỏ phiếu, Alice giữ nguyên → **f = 1**.

Tóm tắt:

- Cho chuỗi **S** (cử tri bỏ phiếu A hoặc B).
- Bạn có thể **thay đổi cử tri** thành **cử tri chiếm đa số** (quyết định theo số phiếu trước đó).
- Tính **số người tối thiểu cần thay đổi để Alice thắng** (hoặc **-1** nếu không thể).



Dịch đề bài:

Bài J: Max Mod

Giới hạn thời gian: 1.5 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Bạn có một **mảng A** gồm **N số nguyên**, ban đầu tất cả các phần tử **$A[i] = 0$** .
Bạn được cho thêm một số nguyên tố **M**.

Có hai loại thao tác cần xử lý:

1. **Cập nhật (Update):** Cho hai số **X** và **Y**, thực hiện:

- **$A[i] += X + (i - 1) * Y$** với **$1 \leq i \leq N$** .

2. **Truy vấn (Query):** Cho hai số **L** và **R**, tìm:

- **Giá trị lớn nhất của $(A[i] \bmod M)$** với **$L \leq i \leq R$** .

Dữ liệu vào:

- Dòng đầu tiên gồm **3 số nguyên: N, M, Q**.
- **Q dòng tiếp theo**, mỗi dòng gồm 3 số nguyên, thể hiện một trong hai loại thao tác:
 - **T = 1, X, Y:** Thao tác **cập nhật**.
 - **T = 2, L, R:** Thao tác **truy vấn**.

Dữ liệu ra:

- Với mỗi **truy vấn**, in ra **giá trị lớn nhất của $(A[i] \bmod M)$** trong đoạn **L đến R**.
-

Ràng buộc:

- $1 \leq N, Q \leq 5 \cdot 10^5$.
 - $1 \leq M \leq 10^9$ (M là số nguyên tố).
 - $1 \leq T \leq 2$.
 - $1 \leq X, Y \leq 10^9$.
 - $1 \leq L \leq R \leq N$.
-

Ví dụ:

Input:

```
5 37 6
1 2 7
1 9 13
2 2 4
1 2 9
2 1 3
2 3 5
```

Output:

```
34
26
35
```

Giải thích:

- **Update 1 (X=2, Y=7):**
 - $A = [2, 9, 16, 23, 30]$.
- **Update 2 (X=9, Y=13):**
 - $A = [2+9, 9+22, 16+35, 23+48, 30+61] \rightarrow A = [11, 31, 51, 71, 91]$.
- **Query 1 (2,4):**
 - $A[2] \bmod 37 = 31 \bmod 37 = 31$.
 - $A[3] \bmod 37 = 51 \bmod 37 = 14$.
 - $A[4] \bmod 37 = 71 \bmod 37 = 34$.
 - Max = **34**.
- **Update 3 (X=2, Y=9):**
 - $A = [11+2, 31+11, 51+20, 71+29, 91+38] \rightarrow A = [13, 42, 71, 100, 129]$.
- **Query 2 (1,3):**
 - $A[1] \bmod 37 = 13 \bmod 37 = 13$.
 - $A[2] \bmod 37 = 42 \bmod 37 = 5$.
 - $A[3] \bmod 37 = 71 \bmod 37 = 34$.
 - Max = **34**.
- **Query 3 (3,5):**
 - $A[3] \bmod 37 = 34$.
 - $A[4] \bmod 37 = 100 \bmod 37 = 26$.
 - $A[5] \bmod 37 = 129 \bmod 37 = 18$.
 - Max = **35**.

Tóm tắt:

- Mảng **A** ban đầu toàn **0**.
- **Cập nhật**: thêm $X + (i-1) * Y$ vào từng phần tử **A[i]**.
- **Truy vấn**: tìm $\max (A[i] \bmod M)$ trên đoạn **[L, R]**.



Dịch đề bài:

Bài K: P thành Q

Giới hạn thời gian: 1 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Bạn được cho hai **hoán vị P và Q** của các số từ **1 đến N**.

- **Điểm số S** ban đầu là **số nghịch thế (inversions)** của hoán vị **P**.
- **Nghịch thế** là số cặp **(i, j)** thỏa mãn $1 \leq i < j \leq N$ và $P[i] > P[j]$.

Bạn có thể thực hiện **tối đa $10 \cdot N$ lần thao tác** sau:

- **Chọn một số nguyên x ($1 \leq x \leq N$):**
 - **Xóa x khỏi P.**
 - **Chèn lại x vào bất kỳ vị trí nào trong P.**
- Sau mỗi thao tác, **cập nhật S**:
 - $S = \max(S, \text{inversions}(P))$.

Mục tiêu:

- **Biến đổi P thành Q.**

- **Tối thiểu hóa giá trị cuối cùng của S** (bạn không cần tối thiểu hóa số thao tác).

Ngoài ra, cần **in ra các thao tác đã thực hiện**.

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T**: số lượng test.
- Mỗi test gồm:
 - Một số nguyên **N**: độ dài hoán vị.
 - Một dòng gồm **N số nguyên**: hoán vị **P**.
 - Một dòng gồm **N số nguyên**: hoán vị **Q**.

Dữ liệu ra:

- Với mỗi test:
 - In ra số nguyên **K** ($0 \leq K \leq 10 \cdot N$) - số thao tác đã thực hiện.
 - **K dòng tiếp theo**, mỗi dòng gồm:
 - **x**: số bị xóa và chèn lại.
 - **pos**: vị trí chèn lại ($1 \leq \text{pos} \leq N$):
 - **pos = i**: chèn **x** ngay trước phần tử thứ **i**.
 - **pos = N**: chèn **x** vào cuối.
-

Ràng buộc:

- $1 \leq T \leq 100$.
- $1 \leq N \leq 500$.

- $1 \leq P[i], Q[i] \leq N$ (không có phần tử trùng lặp).
 - Tổng tất cả N trên các test không vượt quá **500**.
-

Ví dụ:

Input :

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

Output :

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

Giải thích Test 1:

- Ban đầu $P = [1, 2]$, $S = 0$.
- Thao tác: Xóa **1**, chèn vào vị trí 2 $\rightarrow P = [2, 1]$, $S = \max(0, \text{inversions}([2, 1])) = 1$.

- **P** đã thành **Q** = [2, 1].
-

Tóm tắt:

- Bạn được cho hai **hoán vị P và Q**.
- Thực hiện thao tác:
 - **Xóa x khỏi P.**
 - **Chèn lại x vào vị trí bất kỳ.**
- **Cập nhật S** là **ngịch thế lớn nhất** từng đạt được.
- **Mục tiêu:** Biến **P** thành **Q**, đồng thời **tối thiểu hóa giá trị cuối cùng của S**.
- In ra **số thao tác và chi tiết các thao tác**.



Dịch đề bài:

Bài L: Tổng điểm

Giới hạn thời gian: 1 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Cho một **mảng A** gồm **N phần tử**.

Định nghĩa **hàm điểm số (score)** như sau:

- Với một đoạn con **A[L, R] = [A_L, A_{L+1}, ..., A_R]**:
 - **d(L, R)** là **số lượng phần tử khác nhau** trong đoạn con **A[L, R]**.
 - **f(L, R) = (R - L + 1) - d(L, R)**:

- **(R - L + 1)** là độ dài đoạn con.
 - Trừ đi số lượng phần tử khác nhau trong đoạn.
 - Xét **tất cả các cặp (L, R)** sao cho **$1 \leq L \leq R \leq N$** , tìm đoạn con **có giá trị f(L, R) lớn nhất**:
 - Nếu có nhiều đoạn con cùng giá trị **f** lớn nhất, chọn đoạn có **độ dài nhỏ nhất (R - L + 1)**.
 - Nếu vẫn có nhiều đoạn thỏa mãn, chọn bất kỳ.
 - **score(A[L, R])** là **độ dài đoạn con (R - L + 1)** thỏa mãn điều kiện trên.
-

Nhiệm vụ:

- Với **tất cả các đoạn con A[L, R]** trong mảng **A**:
 - Tính **score(A[L, R])**.
 - Tính tổng tất cả các **score** này.
-

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T**: số lượng test.
- Mỗi test gồm:
 - Một số nguyên **N**: độ dài mảng **A**.
 - Một dòng gồm **N số nguyên**: các phần tử **A1, A2, ..., AN**.

Dữ liệu ra:

- Với mỗi test, in ra **tổng score** của tất cả các đoạn con của **A**.

Ràng buộc:

- $1 \leq T \leq 10^4$.
 - $1 \leq N \leq 2 \cdot 10^5$.
 - $1 \leq A[i] \leq N$.
 - Tổng tất cả **N** trên các test **không vượt quá $2 \cdot 10^5$** .
-

Ví dụ:

Input :

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

Output :

```
4
8
26
62
```

Giải thích Test 1:

- Các đoạn con:
 - **[1]** → chỉ có 1 cặp ($L=1, R=1$), **score = 1**.
 - **[1]** → chỉ có 1 cặp ($L=2, R=2$), **score = 1**.
 - **[1,1]** → **d(1,2) = 1** → **f(1,2) = 2-1=1**,
 - **score = 2** (độ dài đoạn con).
 - Tổng **score**: **1 + 1 + 2 = 4**.
-

Tóm tắt:

- Tính **score(A[L, R])** cho tất cả các đoạn con:
 - **f(L, R) = độ dài đoạn con - số lượng phần tử khác nhau**.
 - Chọn đoạn con có **f lớn nhất** và **độ dài nhỏ nhất**.
- **Tổng score** của tất cả các đoạn con.



Dịch đề bài:

Bài M: Tối đa hóa doanh thu bán vé

Giới hạn thời gian: 4 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Có một giải đấu với **128 đội** được đánh số từ **1** đến **128**.

- Với **mỗi cặp ($i < j$)**, đội **j** **mạnh hơn** đội **i** và **luôn thắng** nếu đấu với nhau.
- Đội **i** có **P_i** cổ động viên (không có đội nào chung cổ động viên).

Giải đấu có **7 vòng**. Mỗi vòng:

- Các đội được **chia cặp**, mỗi đội chỉ đấu **1 trận**.
- Đội **thua** bị **loại**, đội **thắng** vào vòng sau.

Cụ thể:

- **Vòng 1:** 128 đội → 64 trận → 64 đội thắng vào vòng sau.
- **Vòng 2:** 64 đội → 32 trận → 32 đội thắng.
- ...
- **Vòng 7:** 2 đội → 1 trận → vô địch.

Giá vé mỗi vòng **tăng dần**:

- **Vòng r:** mỗi vé **giá r**.

Doanh thu của một trận giữa đội **i** và **j** tại vòng **r**:

- $r \times (P_i + P_j)$ (mỗi cổ động viên của hai đội mua vé với giá **r**).

Nhiệm vụ:

- Bạn được phép **chọn cách ghép cặp** các đội ở mỗi vòng.
- Hãy chọn cách ghép sao cho **tổng doanh thu qua các vòng là lớn nhất**.

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T**: số lượng test.

- Mỗi test gồm:
 - Một dòng gồm **128 số nguyên: P1, P2, ..., P128** (số cổ động viên của từng đội).

Dữ liệu ra:

- Với mỗi test, in ra **tổng doanh thu lớn nhất**.

Ràng buộc:

- $1 \leq T \leq 5$.
- $1 \leq P_i \leq 10^9$.

Ví dụ:

Input :

```
2
1 1 1 1 ... (128 lần)
128 127 126 ... 1
```

Output :

```
494
30318
```

Giải thích Test 1:

- Vì tất cả $P_i = 1$, mỗi trận doanh thu là $r \times 2$ (r là số vòng).

- Tổng doanh thu từng vòng:
 - **Vòng 1:** $64 \text{ trận} \times 2 = 128$.
 - **Vòng 2:** $32 \text{ trận} \times 4 = 128$.
 - **Vòng 3:** $16 \text{ trận} \times 6 = 96$.
 - **Vòng 4:** $8 \text{ trận} \times 8 = 64$.
 - **Vòng 5:** $4 \text{ trận} \times 10 = 40$.
 - **Vòng 6:** $2 \text{ trận} \times 12 = 24$.
 - **Vòng 7:** $1 \text{ trận} \times 14 = 14$.
 - Tổng: **494**.
-

Tóm tắt:

- Có **128 đội**, mỗi đội có **P_i** cổ động viên.
- Ghép cặp đấu sao cho **doanh thu tối đa**:
 - Doanh thu trận đấu = **$r \times (P_i + P_j)$** (với **r** là số vòng).
- Chọn ghép cặp qua từng vòng để tối đa hóa **tổng doanh thu**.



Dịch đề bài:

Bài N: Một bài toán cây khung nhỏ nhất nữa

Giới hạn thời gian: 1 giây

Giới hạn bộ nhớ: 1 GB

Mô tả bài toán:

Bạn được cho một **đồ thị vô hướng liên thông G** với:

- **N đỉnh** và **M cạnh**.
- Mỗi cạnh đều có trọng số **1**.

Một số đỉnh được **đánh dấu**, số còn lại **không đánh dấu**.

Bạn được cho một **chuỗi nhị phân S**:

- **S[i] = 1** nếu **đỉnh i** được đánh dấu.
- **S[i] = 0** nếu **đỉnh i** không đánh dấu.

Gọi **X** là tập các đỉnh được đánh dấu.

Bài toán đảm bảo **X luôn khác rỗng**.

Từ tập **X**, xây dựng một **đồ thị đầy đủ H**:

- **Mỗi cặp đỉnh u, v trong X** có một cạnh nối với trọng số là **khoảng cách ngắn nhất giữa u và v trong G** (số cạnh trên đường đi ngắn nhất).

Nhiệm vụ:

- Tính **tổng trọng số của cây khung nhỏ nhất (MST)** trong đồ thị **H**.

Định nghĩa:

- **dist(u, v)**: độ dài đường đi ngắn nhất giữa **u** và **v** trong đồ thị **G** (tính theo số cạnh).

Dữ liệu vào:

- Dòng đầu tiên là số nguyên **T**: số lượng test.
 - Mỗi test gồm:
 - Một dòng gồm **N**, **M**: số đỉnh, số cạnh.
 - Một chuỗi nhị phân **S** độ dài **N**: các đỉnh đánh dấu.
 - **M** dòng, mỗi dòng gồm hai số nguyên **u**, **v**: biểu diễn cạnh **(u, v)** trong đồ thị **G**.
-

Dữ liệu ra:

- Với mỗi test, in ra **tổng trọng số của cây khung nhỏ nhất** trong đồ thị **H**.
-

Ràng buộc:

- $1 \leq T \leq 10^4$.
- $2 \leq N \leq 2 \cdot 10^5$.
- $(N-1) \leq M \leq 2 \cdot 10^5$.
- $S[i] \in \{0,1\}$.
- Luôn tồn tại ít nhất một **i** sao cho **S[i] = 1**.
- $1 \leq u, v \leq N$.
- Các cạnh **(u, v)** là **không trùng lặp**.
- Tổng tất cả **N** và **M** trên các test không vượt quá $2 \cdot 10^5$.

Ví dụ:

Input:

7

3 3

101

1 2

2 3

1 3

6 5

100101

1 2

2 3

3 4

3 5

5 6

4 3

0111

1 2

1 3

1 4

7 7

1110111

1 7

1 4

2 4

3 4

4 5

4 6

4 7

2 1

1 2

6 9

100111

2 5

4 3

3 5

5 1

```
1 6
4 2
4 5
6 3
2 3
6 8
110110
3 5
4 2
2 6
5 1
1 6
6 4
4 3
4 5
```

Output :

```
1
6
4
9
0
3
3
```

Giải thích Test 1:

- Có hai đỉnh được đánh dấu: **1** và **3**.
- **dist(1,3) = 1** (có cạnh trực tiếp).
- Cây khung nhỏ nhất chỉ gồm cạnh đó, tổng trọng số là **1**.

Tóm tắt:

- Cho **đồ thị G** liên thông, các cạnh có trọng số **1**.
- Xác định **tập các đỉnh đánh dấu X**.
- Tạo **đồ thị đầy đủ H** trên **X**, cạnh giữa **u** và **v** có trọng số là **khoảng cách ngắn nhất trong G**.
- Tính **tổng trọng số cây khung nhỏ nhất (MST)** của **H**.