



Bài 1. Chuồng bò [CBRAN] - 6 điểm

Là một người yêu thích kiến trúc hiện đại, nông dân John (FJ) đã xây dựng một chuồng bò mới theo dạng hình tròn hoàn hảo. Bên trong, chuồng bò hình tròn này được chia thành n phòng, đánh số 1, 2, ..., n theo chiều kim đồng hồ quanh chu vi hình tròn. Mỗi phòng đều có hai cửa thông sang phòng bên cạnh cùng với một cửa hướng ra ngoài. FJ có n con bò và anh ta muốn bố trí sao cho mỗi phòng chứa đúng một con bò.

Lý tưởng nhất là khi về chuồng mỗi phòng có đúng một con bò xếp hàng đứng ở cửa hướng ra ngoài. Tuy nhiên chúng thường đi theo đàn và khi về đến chuồng, số bò xếp hàng trước cửa hướng ra ngoài của mỗi phòng là khác nhau. Cụ thể, có c_i con bò đứng xếp hàng ở cửa hướng ra ngoài của phòng i . Tất nhiên $c_i \geq 0 \forall i$ và $\sum_{i=1}^n c_i = n$.

FJ sẽ cho các con bò vào chuồng theo qui tắc sau: Các con bò sẽ đi vào chuồng theo cửa mà nó đứng, sau đó đi vòng quanh các phòng trong chuồng **theo chiều kim đồng hồ**. Phòng trống đầu tiên mà con bò gặp sẽ là phòng nó ngủ qua đêm. Nếu một con bò di chuyển qua d phòng, nó sẽ tiêu thụ một lượng năng lượng là d^2 .

Yêu cầu: Hãy tính tổng năng lượng tối thiểu mà n con bò phải bỏ ra để di chuyển về phòng theo qui tắc mô tả bên trên.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($3 \leq n \leq 10^5$)
- n dòng tiếp theo, dòng thứ i chứa số nguyên dương c_i .

Output:

In ra một số nguyên là tổng năng lượng tối thiểu tìm được.

Example:

Input	Output
10 1 0 0 2 0 0 1 2 2 2	33

Bài 2. Chuồng bò 2 [CBARN2] - 7 điểm

Nông dân John (FJ) xây dựng được một chuồng bò có dạng hình tròn. Bên trong chuồng bò hình tròn này có n phòng đánh số theo chiều kim đồng hồ lần lượt là 1, 2, ..., n . Mỗi phòng đều có hai cửa thông sang phòng bên cạnh và một cửa hướng ra bên ngoài.

FJ muốn bố trí r_i con bò ở phòng i ($i = 1, 2, \dots, n$). Để đàn bò đi vào chuồng một cách có trật tự, FJ chỉ mở k cánh cửa hướng ra bên ngoài đón các con bò về chuồng. Các con bò sau khi vào một trong k cửa sẽ di chuyển **theo chiều kim đồng hồ** đến phòng đầu tiên chưa đủ số bò cần chứa và ở lại phòng này.

Yêu cầu: Hãy giúp FJ chọn ra k cửa hướng ra ngoài để tổng đoạn đường di chuyển của tất cả các con bò sau khi đi vào chuồng là nhỏ nhất. Khi con bò di chuyển đến phòng kế tiếp nó phải di chuyển 1 đơn vị độ dài.

Input:

- Dòng đầu tiên chứa hai số nguyên dương n, k ($3 \leq n \leq 100; 1 \leq k \leq 7$)
- n dòng tiếp theo, dòng thứ i chứa số nguyên dương r_i ($1 \leq r_i \leq 10^6$)

Output:

Một số nguyên duy nhất là kết quả tìm được

Example:

Input	Output
6 2 2 5 4 2 6 2	14

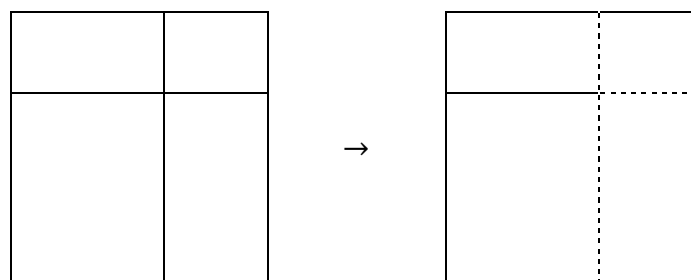
Giải thích: FJ mở cửa ngoài các phòng 2 và 5. Khi đó có 11 con bò đi vào cửa 2 và tổng khoảng cách đi bộ của các con bò này là 8 để vào các chuồng 2, 3, 4. Có 10 con bò đi vào cửa 5 và tổng khoảng cách di chuyển của các con bò này là 6 để vào các chuồng 5, 6 và 1.

Bài 3. Phá tường [BREAKWALL] - 7 điểm

Sàn của một tòa lâu đài rộng có thể được mô tả như là hình chữ nhật với góc dưới-trái là $(0,0)$ và góc trên-phải là (A,B) . Có m bức tường ngang được mô tả như đoạn thẳng nối điểm $(0, b_i)$ với (A, b_i) , ở đây $0 < b_i < B \forall i = 1, 2, \dots, m$. Có n bức tường dọc được mô tả như là các đoạn thẳng nối điểm $(a_i, 0)$ với (a_i, B) ($0 < a_i < A \forall i = 1, 2, \dots, n$). Chu vi của hình chữ nhật cũng là các bức tường. Như vậy tòa lâu đài được chia thành $(m + 1) \times (n + 1)$ phòng.

Người ta muốn phá toàn bộ một số đoạn tường ngang hoặc dọc của một số phòng sao cho tất cả các phòng đều đi đến được nhau. Hãy tính tổng độ dài bé nhất của các đoạn tường cần phá để có thể làm được điều trên?

Ví dụ:



Input:

- Dòng đầu tiên chứa 4 số nguyên dương A, B, n, m ($1 \leq A, B \leq 10^9; n, m \leq 25000$)
- n dòng tiếp theo lần lượt chứa các số nguyên $a_1, a_2, \dots, a_n$ và m dòng cuối cùng lần lượt chứa các số nguyên $b_1, b_2, \dots, b_m$.

Output:

Một số nguyên duy nhất là tổng độ dài nhỏ nhất của các đoạn tường cần phá.

Example:

Input	Output
15 15 5 2 2	44

5	
10	
6	
4	
11	
3	

Subtasks:

- Subtask 1: $n, m \leq 2000$
- Subtask 2: $n, m \leq 25000$

---HẾT---