



På den berömda Påskön står N stycken  uppradade längs en linje som går genom öns mitt.  är numrerade från 0 till $N - 1$.  i (för $0 \leq i < N$) är ursprungligen placerad vid koordinaten $X[i]$. Öns mittpunkt ligger vid koordinaten 0.

Arkitekterna kräver perfekt symmetri i förhållande till mittpunkten. De vill flytta vissa (möjligen inga) av  så att den slutliga placeringen blir symmetrisk runt koordinaten 0. Formellt:

- Varje  måste placeras vid heltalskoordinater.
- Varje heltalskoordinat får innehålla **noll, ett eller flera** .
- För varje heltal $x > 0$ måste antalet  vid koordinaten x vara lika med antalet  vid koordinaten $-x$. Ett godtyckligt antal  (möjligen noll) kan placeras vid koordinaten 0.

M stycken  är dock forntida och för ömtåliga för att flyttas. Deras index är $P[0], P[1], \dots, P[M - 1]$. Dessa M  måste stanna kvar vid sina ursprungliga koordinater. De återstående $N - M$  kan flyttas till valfria heltalskoordinater.

Kostnaden för att flytta en enskild  från koordinaten a till koordinaten b är den absoluta skillnaden $|a - b|$. Den totala kostnaden är summan av kostnaderna för alla  som flyttas.

Din uppgift är att hitta den minsta totalkostnaden för att göra  positioner symmetriska runt koordinaten 0, eller att avgöra att det är omöjligt att uppnå en sådan symmetri under de givna begränsningarna.

Implementationsdetaljer

Du ska implementera följande procedur:

```
long long get_cost(std::vector<int> X, std::vector<int> P)
```

- X : en sorterad lista med längd N som beskriver 🏛️ initiala koordinater.
- P : en sorterad lista med längd M som innehåller indexen för de antika 🗿.
- Denna procedur anropas exakt en gång för varje testfall.

Proceduren ska returnera ett heltal: den minsta totala kostnaden för att göra positionerna symmetriska, eller -1 om det är omöjligt.

Begränsningar

- $1 \leq N \leq 500\,000$
- $0 \leq M \leq N$
- $-10^9 \leq X[0] \leq X[1] \leq \dots \leq X[N-1] \leq 10^9$
- $0 \leq P[0] < P[1] < \dots < P[M-1] < N$

Deluppgifter

Deluppgift	Poäng	Ytterligare begränsningar
1	3	$M = N$
2	4	$M = 0$
3	5	$X[P[j]] < 0$ för varje j så att $0 \leq j < M$.
4	6	$N \leq 10$
5	5	$N \leq 19$
6	5	$N \leq 32$
7	13	$N \leq 200$
8	17	$N \leq 4000$
9	13	$M \leq 4000$
10	29	Inga ytterligare begränsningar.

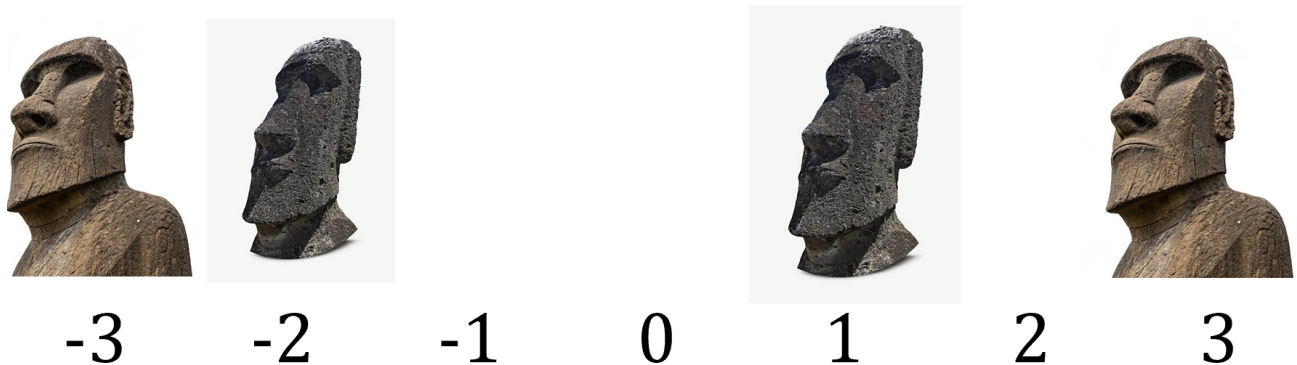
Exempel

Exempel 1

Betrakta följande anrop:

```
get_cost([-3, -2, 1, 3], [1, 2])
```

Den initiala konfigurationen av 🗿 visas i följande figur.



Det finns $N = 4$ 🗿, som initialt är placerade vid koordinaterna $[-3, -2, 1, 3]$. Indexen för de antika 🗿 är $P[0] = 1$ och $P[1] = 2$. Det vill säga, 🗿 vid koordinaterna -2 och 1 kan inte flyttas. De återstående 🗿 med koordinaterna -3 och 3 får flyttas.

För att uppnå symmetri med lägsta möjliga totalkostnad bör vi flytta 🗿 enligt följande:

- Flytta 🗿 från koordinaten -3 till -1 , till en kostnad av $|(-3) - (-1)| = 2$.
- Flytta 🗿 från koordinaten 3 till 2 , till en kostnad av $|3 - 2| = 1$.

Efter dessa flyttningar blir konfigurationen symmetrisk.



Den totala flyttkostnaden är $2 + 1 = 3$, så proceduren bör returnera 3.

Exempel 2

Betrakta följande anrop:

```
get_cost([2, 2, 2, 3], [])
```

Här gäller $M = 0$, vilket innebär att inga 🗿 är forntida och att alla 🗿 kan flyttas. En lösning med lägsta totalkostnad är att flytta alla 🗿 till koordinaten 0. Kostnaden för varje 🗿 är:

- $|2 - 0| = 2$ för 🗿 0, 1 och 2.
- $|3 - 0| = 3$ för 🗿 3.

Den totala kostnaden är $2 + 2 + 2 + 3 = 9$. Därför bör proceduren returnera 9. Observera att det finns andra lösningar med samma totala kostnad.

Exempel 3

Betrakta följande anrop:

```
get_cost([1, 2, 3, 4], [0, 1, 2, 3])
```

Alla 4 🗿 är forntida och kan inte flyttas. Eftersom deras koordinater alla är positiva är det omöjligt att göra konfigurationen symmetrisk runt 0. Därför bör proceduren returnera -1 .

Exempel på exempeldomaren

Indataformat:

```
N M
X[0] X[1] ... X[N-1]
P[0] P[1] ... P[M-1]
```

Observera att den tredje raden kan vara tom om M är 0.

Utdataformat:

```
C
```

Här är C det värde som returneras av `get_cost`.