

Problem - C: Layered Cake

- **Question Analysis:**

Chef 2 Layer এর cake বানাতে।

- 1st Layer এর জন্য N সংখ্যক option আছে। যেখানে A_i এদের Size নির্দেশ করে।
- 2nd Layer এর জন্য M সংখ্যক option আছে। যেখানে B_i এদের Size নির্দেশ করে।

তাকে এমনভাবে 2 layer এর cake বানাতে হবে যাতে - 2nd layer < 1st layer
মোট কতগুলো এই condition maintain করে cake বানাতে পারবে, সেটা count করতে হবে।

- **Hints:**

- N, M - এর constraint খুবই কম। $N, M \leq 100$
- Brute force ভাবে চিন্তা কর।
- তোমাদের Week-8, sorting & searching এর কথা চিন্তা কর।

Bonus [Must Try , if you get AC or Not. Again Try it]

- **Constraint:**

$$1 \leq T \leq 10^4$$

$$1 \leq N, M \leq 10^5$$

$$1 \leq A_i, B_i \leq 10^9$$

It is guaranteed that N, M over all test cases does not exceed 10^5 .

Problem - D: Sorting Prefixes

- **Question Analysis:**

আমাকে একটা **array A** দেওয়া আছে, যেটা মূলত একটা **permutation [1, N]**

Operation:

- একটা **index i choose** কর। তারপর **Ai cost** এর বিনিময়ে **1** থেকে **i** **index** পর্যন্ত **element** - গুলোকে **sort** করা।

আমার কাজ হচ্ছে - **operation** টি ইচ্ছামতো **use** করে, পুরো **array** কে **sort** করা এবং এমনভাবে করা যাতে **total cost minimize** হয়।

- **Hints:**

- **Sample Test cases** সবগুলো **Dry Run** কর।
 - **Last element** থেকে চিন্তা কর।
 - **Complex** কিছু না, **just observation**,
 - **Last** কোন **element** টা অবশ্যই **choose** করতে হবে যাতে **sort** হয়।
 - **i th element** যদি **Choose** করি, তাহলে **i-1 th** , **i-2 th** , So on. **Element** গুলো কি আর **choose** করার দরকার আছে?
 - **If Yes, Why?**
 - **If No, also Why?**
-

Problem - E: Sell Upto 2

- **Question Analysis:**

আমার কাছে **N days** এবং **i th day** তে **watch sell** করার **cost** দেওয়া আছে।

এখানে প্রতিদিন ২ টা ঘটনা ঘটে।

- ১ টা **watch free** তে পাব
- এখন এই **watch** আমাকে বর্তমান অথবা ভবিষ্যতের কোন দিন **sell** করার সুযোগ আছে। কিন্তু আমি কোনোদিন ২ টার বেশি **sell** করতে পারব না।

আমার **Target** হচ্ছে **maximum watch sell** করে , বেশি বেশি **profit** করা।

- **Hints:**

- যেহেতু **maximum profit** লাগবে, তাই সব **watch sell** করতে হবে।
- শেষের দিন থেকে ভাব।
- প্রতি **i th day** তে কিন্তু তোমার ২ টা **option** , হয় এখন **sell** , নাহলে **i+1 th** থেকে **n th day** অবধি যেকোনো একদিন **sell**

- যদি **future day** তে বেশি দামে বিক্রি করতে চায়, **Future** এর **value** গুলোর **maximum value track** রাখ।

- **Multiset, priority queue, basic sorting** - এগুলো **use** করতে পার।
- **Sample testcase** এর -

4

2 7 3 5

এটা **Dry Run** কর, পিছন থেকে, মনে কর, ওই দিনের **value** টা তুমি ২ টা **watch sell** করার জন্যই **Use** করবে,
