Editorial: Ticket Hoarding - 1951C

Avighna

June 2024

There is but one observation you need to make to solve this problem, and I'll prove this immediately after stating it: the order of the days in which you buy the tickets doesn't change the answer.

Slightly more formally, buying a tickets at a cost of u and then buying b tickets at a cost of v will require the same amount of money as buying b tickets at a cost of v and then buying a tickets at a cost of u. The price we pay for the tickets is the price of a ticket added with the number of tickets we've already bought, multiplied by the number of tickets we're buying.

Let the number of tickets we've bought before this whole transaction be k.

In the first case, the price we pay is $(a+k)u+(b+k+u)v = au+ku+bv+kv+uv$. In the second case, it's $(b+k)v+(a+k+v)u = bv+kv+au+ku+uv$. Both expressions are equal, so the order in which we buy tickets doesn't actually matter.

This justifies us being able to reorder the prices array however we want. Sorting it ascendingly is most convenient, since we can buy as many tickets as possible (without actually exceeding how many we need) on prior days as opposed to subsequent days, where they'll be more expensive.

1 Code

```
void solve() {
    ll n, m, k;
    cin >> n >> m >> k;
    vector<ll> price(n);
    for (auto &i : price) {
        cin >> i;
    }
    sort(price.begin(), price.end());

    ll ans = 0;
    ll bought = 0;
    for (ll i = 0; i < n; ++ i) {
        ll x = min(m, k);
        ans += (price[i] + bought) * x;
        bought += x, k -= x;
    }

    cout << ans << "\n";
}
```