

Minimum product subset of an array

maxNeg记录最大的负数

minPos记录最小的正数

negCount记录负数的个数

zeroCount记录0的个数

```
#include <bits/stdc++.h>

using namespace std;

int minProd(vector<int>& arr) {
    sort(arr.begin(), arr.end());
    int prod = 1;
    int negCount = 0, zeroCount = 0;
    int maxNeg = INT_MIN, minPos = INT_MAX;
    for (int num: arr) {
        if (num == 0) {
            zeroCount++;
            continue;
        }
        if (num < 0) {
            negCount++;
            maxNeg = max(maxNeg, num);
        }
        if (num > 0) {
            minPos = min(minPos, num);
        }
        prod *= num;
    }
    // 如果数组为全0, 或者数组中没有负数且数组中有0, 把么最小积是0
    if (zeroCount == arr.size() || (zeroCount > 0 && negCount == 0)) return 0;
    // 如果数组中没有负数也没有0, 那么最小积是 最小的正数
    if (negCount == 0 && zeroCount == 0) return minPos;
    // 如果数组中负数的个数为偶数且数组中没有0, 那么最小积是所有数乘积然后除以最大的负数
    if (negCount / 2 == 0 && zeroCount == 0) return prod / maxNeg;
    return prod;
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        scanf("%d", &N);
        vector<int> arr;
        for (int i = 0; i < N; ++i) {
```

```
        int num;
        scanf("%d", &num);
        arr.push_back(num);
    }
    printf("%d\n", minProd(arr));
}
return 0;
}
```