Minimum Cost of ropes

> 给定N段绳子，将这N段绳子合并为一条绳子，合并两条绳子的cost是两条绳子长度之和，求合并所有绳子最小的cost

类似于huffman树的构造，采用优先级队列即可

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

using namespace std;

typedef vector<long long int> vl;

long long int minCostRopes(vl &ropes, int &N) {
    long long int minCost = 0;
    priority_queue<long long int, vl, greater<long long int>> q;
    for (auto rope: ropes) q.push(rope);
    while (!q.empty() && q.size() >= 2) {
        long long int tempCost = q.top();
        q.pop();
        tempCost += q.top();
        q.pop();
        minCost += tempCost;
        q.push(tempCost);
    }
    return minCost;
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        scanf("%d", &N);
        vl ropes(N);
        for (int i = 0; i < N; ++i) scanf("%lld", &ropes[i]);
        printf("%lld\n", minCostRopes(ropes, N));
    }
}
```