

1232 Crossing River

题目描述

几个人过河，每次过两人一人回，速度由慢者决定，问过河所需最短时间。

输入

输入 t 组数据，每组数据第1行输入 n ，第2行输入 n 个数，表示每个人过河的时间。

输出

输出 t 行数据，每行1个数，表示每组过河最少时间。

输入样例

```
1
4
1 2 5 10
```

输出样例

```
17
```

解析

分情况讨论

只剩下1人，总用时就是自己，即 T_0

T_0



剩下2人，总用时是最多的人，即 T_1

T_0



T_1



剩下3人，先用最快的人送走最慢的人，再回来，然后一趟就过去了，总用时是 $T_2 + T_1 + T_0$

第一次送人所用时间

T_0



T_2



返回时间

T_0



第二次送人所用时间

T0 

T1 

大于等于四人时，存在两种方案，需要进行比较。

方案一：始终用最快的人送最慢的人，总用时是 $T_3 + T_2 + 2 \times T_0$

第一次送人所用时间

T0 

T3 

第一次返回时间

T0 

第二次送人所用时间

T0 

T2 

第二次返回时间

T0 

方案二：用速度最快的和次快的循环载人，即最快的和次快的先过河，次快的留下，最快的回来，接着最慢和次慢的人过河，次快的回来。总用时： $2 \times T_1 + T_0 + T_3$

第一次送人所用时间

T0 

T1 

第一次返回时间

T0 

第二次送人所用时间

T2 

T3 

第二次返回时间

T1

最后比较两个方案中最小值即可。

编码

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

#define MAX 1005
using namespace std;

int main() {
    //测试用例的数量
    int caseNum;
    //每次过河的人数，和索引
    int n, i;
    int person[MAX];
    cin >> caseNum;
    while (caseNum--) {
        cin >> n;
        for (i = 0; i < n; i++) {
            cin >> person[i];
        }
        //排序每个人的过河时间，升序排列
        sort(person, person + n);
        //重置总的过河时间
        int sum = 0;
        //四人以上，需要进行动态规划
        //一次通过两人，因此是i=i-2
        for (i = n - 1; i > 2; i = i - 2) {
            //策略1:最快的载人
            int tmp1 = 2 * person[0] + person[i] + person[i - 1];
            //策略2:最快的和次快的循环载人
            int tmp2 = person[0] + 2 * person[1] + person[i];
            sum += min(tmp1, tmp2);
        }
        //剩下三个人
        //先用最快的人送走最慢的人，再回来，然后一趟就过去了
        //最慢的人的速度是: person[2]
        //自己回来是: person[0]
        //最后一起过去的时间是: person[1]
        if (i == 2) {
            sum += person[0] + person[1] + person[2];
        }
        //剩下两个人，直接一趟过去，选择速度最慢的值
        if (i == 1) {
            sum += person[1];
        }
        //剩下一个人，直接自己过去
    }
}
```

```
    if (i == 0) {  
        sum += person[0];  
    }  
    cout << sum << endl;  
}  
  
return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

