

2022 10 29 div2+div1

Bricks and Bags

将若干个数字分成三组。
每一组保证有1个数字。
分配完成之后，每一组选择一个数字。
标记为 w_1, w_2, w_3 。
求解 $|w_1 - w_2| + |w_2 - w_3|$ 的最小值；

thinking

- 做题时，一直想着贪心。
 - 贪心如下情况
 - 最大放中间
 - 最小放中间
 - 次大放中间。
 - 次小放中间。
 - 两个最大放两间。
 - 两个最小放两边。
 - 但是都没有过。
- 正解
 - 首先将数组排一个序。
 - 通过

$$p_1 < p_2 < p_3$$

$$p_1 > p_2 > p_3$$

$$p_2 < \min(p_1, p_3) \quad \text{and} \quad \min(p_1, p_3) = p_2 + 1$$

$$p_2 > \max(p_1, p_3) \quad \text{and} \quad \max(p_1, p_3) = p_2 - 1$$

- 很难发现，可以构造出最终选择上方的任何形式。
 - 像第一第二种又怎么理解？
 - 对于第一种第二种，最大的方案总是 $mmax - mmin$ 。
 - 但是它肯定不是最大的，当最小的放在中间的是时候，最大放在一边，其它全放在另一端，
 - 总有更大的情况。
 - 如果第一第二种的情况，是否必然比第三第四种的情况小？
 - 第三种，取中间的 a_i ，比他小的全部放中间，最大的放左边，比它大（除了最大）的放右边
 - 第四种，比它小的除了最小全部放左边，最下放右边，比它大的放中间。
- 问题转化成找上方的形式的方案中的最大值。

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;

const int maxn = 2e5 + 10;
```

```

11 a[maxn];

void solve()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
        cin >> a[i];
    sort(a + 1, a + 1 + n);
    ll ans = a[n] - a[1];
    a[n + 1] = a[n];
    a[0] = a[1];
    for (int i = 1; i <= n; i++)
        ans = max({ans, a[n] - a[i] + a[i + 1] - a[i], a[i] - a[1] + a[i] - a[i
- 1]});
    cout << ans << '\n';
}

int main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr), cout.tie(nullptr);
    int t;
    cin >> t;
    while (t--)
        solve();
}

```