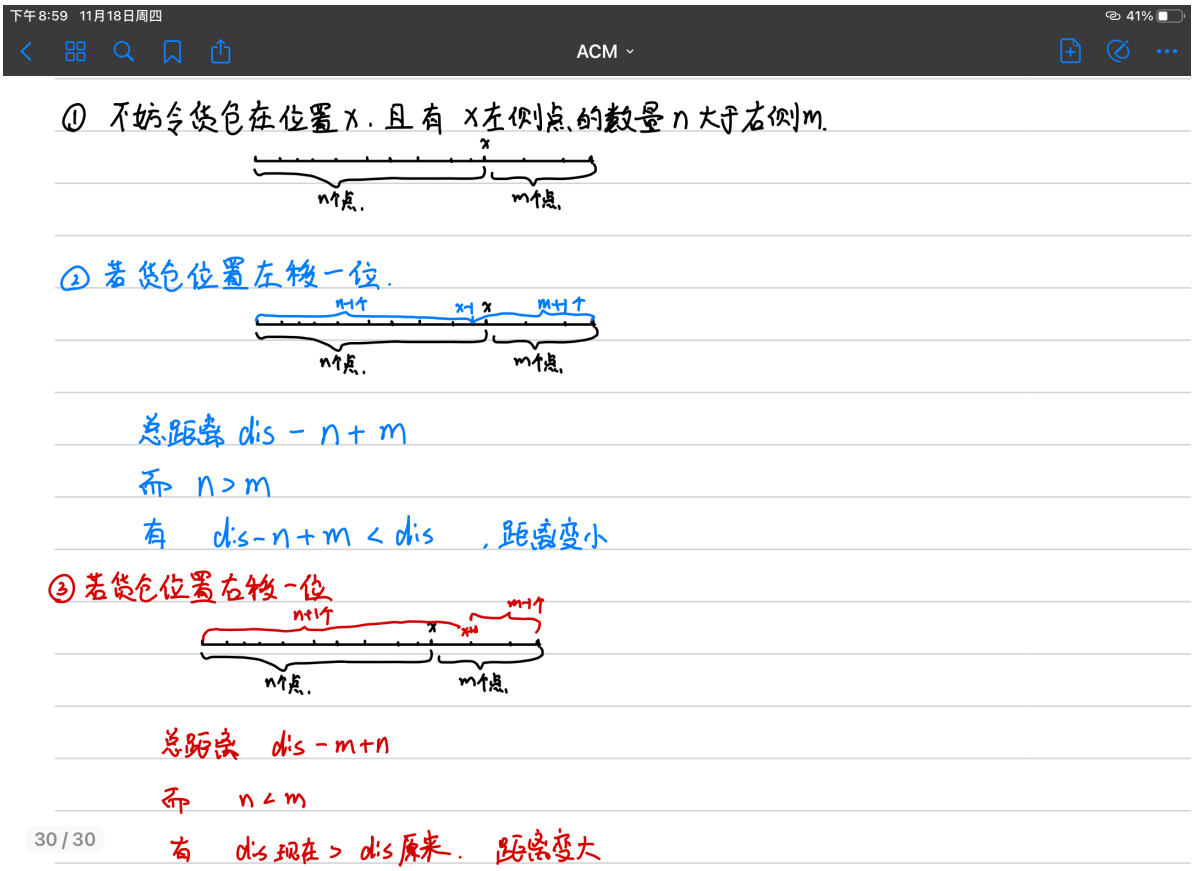
中位数问题

acwing 104 货仓选址

思路

取坐标的中位数，作为仓库的位置。

证明



代码

```cpp
#include <bits/stdc++.h>
#define endl "\n"

using namespace std;

const int maxn = 100010;
int a[maxn];
int res;

int main(){
    ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
    int n; cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> a[i];
    }

    sort(a + 1, a + 1 + n);
```

```
    int mid = 1 + n >> 1;

    for(int i = 1; i <= n; i++){
        res += abs(a[mid] - a[i]);
    }

    cout << res << endl;

    return 0;
}
```

acwing 1536 均分纸牌

思路

如果有解 -> 总牌数能被整除

从最左边开始

若第一堆牌的数量不足平均数 -> 向第二堆借相应的牌数,直到平均数

若第一堆牌的数量大于平均数 -> 给第二堆牌相应的牌数,直到平均数

...

以此类推,直到最后一堆牌, 其数量也一定是平均数

代码

```
#include <bits/stdc++.h>
#define endl "\n"

using namespace std;

const int maxn = 110;
int n, sum, ans;
int a[maxn];

int main(){
    ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
    cin >> n;

    for(int i = 1; i <= n; i++){
        cin >> a[i];
        sum += a[i];
    }

    int avg = sum / n;

    for(int i = 1; i <= n; i++){
        if(a[i] != avg) a[i + 1] += a[i] - avg, ans ++;
    }

    cout << ans << endl;
    return 0;
}
```

acwing 1208 翻硬币 (非中位数问题,但与上题贪心思想类似)

思路

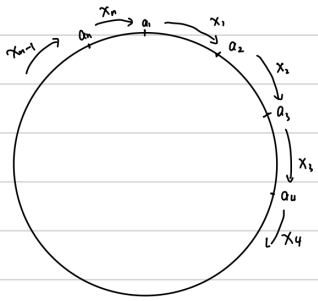
对一个字符串，从前向后遍历。如有不一致，则将该位置与该位置之后以为的字符反转。

代码

```cpp
#include <bits/stdc++.h>
#define endl "\n"

using namespace std;

string s1, s2;
int cnt;

char change(char s){
    if(s == 'o') return '*';
    else return 'o';
}

int main(){
    ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);

    cin >> s1 >> s2;

    for(int i = 0; i < s1.length(); i++){
        if(s1[i] != s2[i]){
            cnt ++;
            s1[i] = change(s1[i]);
            s1[i + 1] = change(s1[i + 1]);
        }
    }

    cout << cnt;

    return 0;
}
```

acwing 122 糖果传递

思路 + 证明



$b=\frac{1}{n}(a_1+a_2+\cdots+a_n)$

目标：最小化 $|x_1|+|x_2|+|x_3|+\cdots+|x_n|$

$$\begin{cases} a_1-x_1+x_n=b \\ a_2-x_2+x_1=b \\ \vdots \\ a_n-x_n+x_{n-1}=b \end{cases} \Rightarrow \begin{cases} x_1=x_n-(b-a_1) \\ x_2=x_1-(b-a_2)=x_1-(2b-a_1-a_2) \\ \vdots \qquad \vdots \\ x_n=x_n-(nb-a_1-a_2-\cdots-a_{n-1}) \end{cases}$$

$$\text{原式}=|x_n-(b-a_1)|+|x_n-(2b-a_1-a_2)|+\cdots+|x_n-a|$$

$$=|x-c_0|+|x-c_1|+|x-c_2|+\cdots+|x-c_{n-1}|$$

转化为货仓选址.

$\Rightarrow$ x 取中位数.

代码

```cpp
#include <bits/stdc++.h>
#define endl "\n"
typedef long long ll;
using namespace std;
const int maxn = 1e6 + 10;

ll a[maxn], s[maxn], c[maxn];
int n, ans;
ll sum;

int main(){
    ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> a[i];
        s[i] = s[i - 1] + a[i];
    }

    ll avg = s[n] / n;

    for(int i = 1; i <= n; i++){
        c[i - 1] = i * avg - s[i];
    }


    nth_element(c, c + n / 2, c + n);
    ll res = 0;
    for(int i = 0; i < n; i++){
        res += abs(c[i] - c[n / 2]);
    }

    cout << res;
    return 0;
}

```

acwing 105 七夕祭

思路

对于每一行的元素，我们可以求出他们的行和。而在行和中首尾元素相连，这就转化为糖果传递问题。

又因为，我们一次移动两个数，仅仅会影响其单一行或单一列的目标，而不会同时影响一行和一列，所以我们单独处理两个。如果合法，那么就两个答案加起来，如果单个合法，那么就处理单个就好了。

代码

```cpp
#include <bits/stdc++.h>
#define endl "\n"
typedef long long ll;

using namespace std;

const int maxn = 1e5 + 10;
int n, m;
int row[maxn], col[maxn], s[maxn], c[maxn];

ll work(int a[], int n){ //环形均分糖果问题
    for(int i = 1; i <= n; i++){
        s[i] = s[i - 1] + a[i];
    }

    if(s[n] % n) return -1;

    int avg = s[n] / n;

    c[1] = 0;
    for(int i = 2; i <= n; i++){
        c[i] = s[i - 1] - (i - 1) * avg;
    }

    sort(c + 1, c + n + 1);
    ll res = 0;
    for(int i = 1; i <= n; i++) res += abs(c[i] - c[(n + 1) / 2]);

    return res;
}

int main(){
    ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);

    int t; cin >> n >> m >> t;

    while(t--){
        int x, y; cin >> x >> y;
        row[x] ++, col[y] ++;
    }

    ll r = work(row, n);
    ll c = work(col, m);

```

```
45      if(r != -1 && c != -1) cout << "both " << r + c;
46      else if(r != -1) cout << "row " << r;
47      else if(c != -1) cout << "column " << c;
48      else cout << "impossible";
49
50      return 0;
51  }
```

acwing 106 动态中位数

思路

代码