

4.10 课堂笔记 6

本周作业:

- 1、一维前缀和: P3662
- 2、二分: P4058

上周作业答案:

P6568

```
1  #include <iostream>
2  using namespace std;
3  int a[1000005], sum[1000005], n, k, t;
4  int main()
5  {
6      cin >> n >> k;
7      for(int i=1; i<=n; i++){
8          cin >> a[i];
9          sum[i] = sum[i-1] + a[i];
10     }
11     int ans = 0;
12     for(int i=1; i<=n-k; i++){
13         t = sum[i+k] - sum[i-1];
14         ans = max(ans, t);
15     }
16     cout << ans;
17     return 0;
18 }
```

P3406

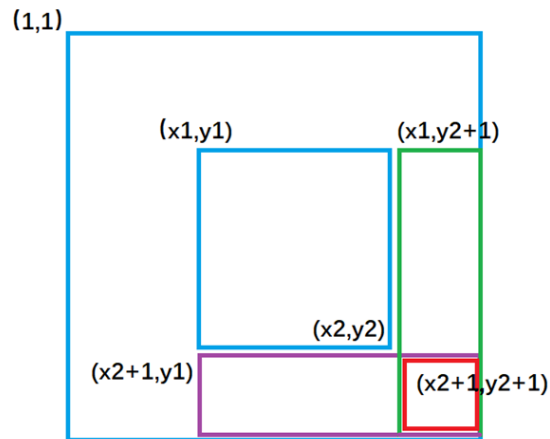
```
4  int _n, _m;
5  long long _cha[100010];
6  long long _list[100010][4];
7  long long _res;
8  void ChaFen(int a, int b) {
9      if(a > b) {
10         swap(a, b);
11     }
12     _cha[a]++;
13     _cha[b]--;
14 }
15 int main () {
16     cin >> _n >> _m;
17     int begin = 0;
18     int end = 0;
19     cin >> begin;
20     for(int i = 2; i <= _m; i++) {
21         cin >> end;
22         ChaFen(begin, end);
23         begin = end;
24     }
25     for(int i = 1; i <= _n - 1; i++) {
26         cin >> _list[i][1] >> _list[i][2] >> _list[i][3];
27         _cha[i] += _cha[i - 1]; //表示每个区间经过的次数
28         _res += min(_list[i][3] + _cha[i] * _list[i][2], _list[i][1] * _cha[i]); //比较每个区间两种方案的花费
29     }
30     cout << _res;
31 }
```

二维差分

$a[i][j]$ 数组是 $b[i][j]$ 数组的前缀和数组，那么 $b[i][j]$ 是 $a[i][j]$ 的差分数组

原数组: $a[i][j]$

构造差分数组: $b[i][j]$



如何将原数组 $(x1, y1)$ 到 $(x2, y2)$ 的矩阵中的每个元素值都加上 c ?

对差分数组进行以下操作:

$b[x1][y1] += c;$

$b[x1][y2+1] -= c;$

$b[x2+1][y1] -= c;$

$b[x2+1][y2+1] += c;$

练习 1: 二维差分

P3397

```
1  #include<iostream>
2  using namespace std;
3  int n,m;
4  int x1, y1, x2, y2;
5  int mapp[1010][1010],c[1010][1010];
6  int main(){
7      cin >> n >> m;
8      for(int i = 1; i <= m; i++) { //差分数组
9          cin >> x1 >> y1 >> x2 >> y2;
10         c[x1][y1] += 1;
11         c[x1][y2 + 1] -= 1;
12         c[x2 + 1][y1] -= 1;
13         c[x2 + 1][y2 + 1] += 1;
14     }
15     //求各点地毯数量
16     for(int i = 1; i <= n; i++) {
17         for(int j = 1; j <= n; j++) {
18             mapp[i][j] = mapp[i - 1][j] + mapp[i][j - 1] - mapp[i - 1][j - 1] + c[i][j]; //前缀和
19         }
20     }
21     for(int i = 1; i <= n; i++) {
22         for(int j = 1; j <= n; j++) {
23             cout << mapp[i][j] << " ";
24         }
25         cout << endl;
26     }
27     return 0;
28 }
```

练习 2：一维差分综合题

P2671

推导：

```

1  一共有2m组，现在分析其中一组的情况：
2  设一个分组里有k个数，这个分组中的数值分别是num[1],num[2]... num[k]，下标分别是xbiao[1],xbiao[2]... xbiao[k]
3  那么可得这一组的值为：
4
5  = (num[1]+num[2])*(xbiao[1]+xbiao[2]) + (num[1]+num[3])*(xbiao[1]+xbiao[3])+... +(num[1]+num[k])*(xbiao[1]+xbiao[k])
6
7  +(num[2]+num[3])*(xbiao[2]+xbiao[3])+(num[2]+num[4])*(xbiao[2]+xbiao[4])+... +(num[2]+num[k])*(xbiao[2]+xbiao[k])
8
9  +... ..
10
11 +(num[k-1]+num[k])*(xbiao[k-1]+xbiao[k])
12
13
14 =num[1]*(xbiao[1]+xbiao[2]+xbiao[1]+xbiao[3]+... +xbiao[1]+xbiao[k]) //k-1个 xbiao[1]
15
16 +num[2]*(xbiao[1]+xbiao[2]+xbiao[2]+xbiao[3]+... +xbiao[2]+xbiao[k])
17
18 +... ..
19
20 +num[k]*(xbiao[1]+xbiao[k]+xbiao[2]+xbiao[k]+... +xbiao[k-1]+xbiao[k])
21
22
23 =num[1]*(xbiao[1]*(k-2)+xbiao[1]+xbiao[2]+... +xbiao[k])
24
25 +num[2]*(xbiao[2]*(k-2)+xbiao[1]+xbiao[2]+... +xbiao[k])
26
27 +... ..
28
29 +num[k]*(xbiao[k]*(k-2)+xbiao[1]+xbiao[2]+... +xbiao[k])//前缀和
30
31 即得“相同颜色+下标奇偶相同”这一组的下标前缀和： xbiao[1]+xbiao[2]+... +xbiao[k]
32 分成2m组进行前缀和预处理，并分成2m组进行计算

```

```

7  {
8      cin >> n >> m;
9      for(int i=1;i<=n;i++){
10         cin >> num[i];
11     }
12     for(int i=1;i<=n;i++){
13         cin >> color[i];
14     }
15     for(int i=1;i<=n;i++){
16         if(i%2==0){
17             tot[color[i]]++;//记录“偶数下标+某个颜色”那一段的个数
18             sum[color[i]] = (sum[color[i]]+i)%mod;//预处理前缀和：下标前缀和
19         }
20         else{
21             tot[color[i]+m]++;//+m防止重复。记录“奇数下标+某个颜色”那一段的个数
22             sum[color[i]+m]=(sum[color[i]+m]+i)%mod;
23         }
24     }
25     for(int i=1;i<=n;i++){
26         if(i%2==0){
27             ans=(ans+num[i]*(i*(tot[color[i]]-2)%mod+sum[color[i]]))%mod;//套用公式
28         }
29         else{
30             ans=(ans+num[i]*(i*(tot[color[i]+m]-2)%mod+sum[color[i]+m]))%mod;
31         }
32     }
33     cout << ans;

```

练习 3: 二分答案+前缀和 综合题

P1314

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define maxn 200005
4  #define ll long long
5  int n,m;
6  ll S,a;
7  int w[maxn],v[maxn],l[maxn],r[maxn];
8  ll sum[maxn],cnt[maxn];
9
10 bool judge(int W){
11     // memset(sum,0,sizeof(sum)); //数组清零
12     // memset(cnt,0,sizeof(cnt));
13     for(int i=1;i<=n;i++){
14         if(w[i]>=W){
15             sum[i] = sum[i-1] + v[i]; //矿石的重量大于参考值, 我们才要求它的价值之和
16             cnt[i] = cnt[i-1] + 1;
17         }
18         else{
19             sum[i] = sum[i-1]; //矿石的重量小于参考值
20             cnt[i] = cnt[i-1];
21         }
22     }
23     ll tmp=0;
24     for(int i=1;i<=m;i++){
25         tmp+=(cnt[r[i]]-cnt[l[i]-1])*(sum[r[i]]-sum[l[i]-1]); //tmp就是检验值Y
26     }
27     a = llabs(S-tmp); //保存 |S-tmp|
28     if(tmp < S) return true;
29     else return false;
30 }
31
32
33 int main(){
34     cin >> n >> m >> S;
35     int mx=0;
36     for(int i=1;i<=n;i++){
37         cin >> w[i] >> v[i];
38         mx=max(w[i],mx); //保存最大重量
39     }
40     for(int i=1;i<=m;i++){
41         cin >> l[i] >> r[i];
42     }
43     int l=0,r=mx+1; //矿石全选和全不选的情况
44     ll ans=0x3f3f3f3f3f3f3f3f; //long long 类型的无穷大
45     while(l<=r){
46         int mid=(l+r)/2; //更新参考重量W
47         if(judge(mid))r=mid-1; //注意二分的取值
48         else l=mid+1;
49         ans=min(ans,a); //更新|S-tmp|的最小值
50     }
51     cout << ans;
52     return 0;
53 }

```