

数位dp

不含前导零且相邻两个数字之差至少为 2 的正整数被称为 windy 数。windy 想知道，在 a 和 b^* 之间，包括 a 和 b ，总共有多少个 windy 数？

```
#include<bits/stdc++.h>
using namespace std;
//设dp[i][j]为长度为i中最高位是j的windy数的个数
//方程 dp[i][j]=sum(dp[i-1][k]) 其中 abs(j-k)>=2
int p,q,dp[15][15],a[15];
void init()
{
    for(int i=0;i<=9;i++) dp[1][i]=1;
    for(int i=2;i<=10;i++)
    {
        for(int j=0;j<=9;j++)
        {
            for(int k=0;k<=9;k++)
            {
                if(abs(j-k)>=2)
                    dp[i][j]+=dp[i-1][k];
            }
        }
    }
}
int work(int x)
{
    memset(a,0,sizeof(a));
    int len=0,ans=0;
    while(x)
    {
        len++;
        a[len]=x%10;
        x/=10;
    }
    //分为几个板块 先求len-1位的windy数 必定包含在区间里的
    for(int i=1;i<=len-1;i++)
    {
        for(int j=1;j<=9;j++)
        {
            ans+=dp[i][j];
        }
    }
    //然后是len位 但最高位<a[len]的windy数 也包含在区间里
    for(int i=1;i<a[len];i++)
    {
        ans+=dp[len][i];
    }
    //接着是len位 最高位与原数相同的 最难搞的一部分
    for(int i=len-1;i>=1;i--)
```

```

{
    //i从最高位后开始枚举
    for(int j=0;j<=a[i]-1;j++)
    {
        //j是i位上的数
        if(abs(j-a[i+1])>=2)
            ans+=dp[i][j]; //判断和上一位(i+1)相差2以上
            //如果是 ans就累加
    }
    if(abs(a[i+1]-a[i])<2) break;
    // if(i==1) ans+=1;
}
return ans;
}
int main()
{
    init();
    cin>>p>>q;
    cout<<work(q+1)-work(p)<<endl;
    return 0;
}

```

背包

1.01背包

(1) 基础

(2) 前k个最优解

```

#include <bits/stdc++.h>
using namespace std;
#define ll long long
const int inf=0x3f3f3f3f;
const int N=5e3+1;
struct node
{
    int weight;
    int value;
};
node a[N];
int cmp[N];
int dp[N][N];
void solve()
{
    int k,v,n;
    cin>>k>>v>>n;
    for(int i=1;i<=n;i++)

```

```

{
    cin>>a[i].weight>>a[i].value;
}
//01 bag
// for(int i=1;i<=n;i++)
// {
//     for(int j=v;j>=0;j--)
//     {
//         if(j-a[i].weight<0)continue;
//         dp[j]=max(dp[j],dp[j-a[i].weight]+a[i].value);
//     }
// }
memset(dp,~0x3f,sizeof(dp));

dp[0][1]=0;
for(int i=1;i<=n;i++)
{
    for(int j=v;j>=a[i].weight;j--)
    {
        int pos=1;
        int x=1;int y=1;
        while(pos<=k)
        {
            if(dp[j][x]>dp[j-a[i].weight][y]+a[i].value)
            {
                cmp[pos]=dp[j][x];
                x++;
            }
            else
            {
                cmp[pos]=dp[j-a[i].weight][y]+a[i].value;
                y++;
            }
            pos++;
        }
        for(int t=1;t<=k;t++)
        {
            dp[j][t]=cmp[t];
        }
    }
}
int ans=0;
for(int i=1;i<=k;i++)
{
    ans+=dp[v][i];
}
cout<<ans;
return ;
}
int main()
{
    ios::sync_with_stdio(false);

```

```

    cin.tie(0);cout.tie(0);
    solve();
    return 0;
}

```

二维优化

[E - Maximum Monogonosity](#)

```

#include <bits/stdc++.h>
using namespace std;
#define ll long long
constexpr ll inf=1e18;
constexpr ll N=2e6+10;
constexpr ll M=3e3+10;

ll a[N];
ll b[N];
void solve()
{
    ll n,k;
    cin>>n>>k;
    vector<vector<ll>> dp;
    dp.assign(n + 1, vector<ll>(k + 1, 0));

    vector<ll>min1;
    vector<ll>min2;
    vector<ll>max1;
    vector<ll>max2;

    min1.assign(n+1, inf);
    min2.assign(n+1, inf);
    max1.assign(n+1,-inf);
    max2.assign(n+1,-inf);

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

    for(ll i=0;i<=n;i++)
    {
        for(ll j=0;j<=min(k,i);j++)
        {
            if(i!=0)dp[i][j]=dp[i-1][j];

```

```

        if(i!=0)
        {
            dp[i][j]=max(dp[i][j],-a[i]+b[i]-min1[i-j]);//- + - +
            dp[i][j]=max(dp[i][j],+a[i]+b[i]-min2[i-j]);//+ + - -
            dp[i][j]=max(dp[i][j], a[i]-b[i]+max1[i-j]);//+ - + -
            dp[i][j]=max(dp[i][j],-a[i]-b[i]+max2[i-j]);//- - + +
        }
        if(i+1<=n)
        {
            min1[i-j]=min(a[i+1]-b[i+1]-dp[i][j],min1[i-j]);
            min2[i-j]=min(a[i+1]+b[i+1]-dp[i][j],min2[i-j]);
            max1[i-j]=max(a[i+1]-b[i+1]+dp[i][j],max1[i-j]);
            max2[i-j]=max(a[i+1]+b[i+1]+dp[i][j],max2[i-j]);
        }

    }
}
// cout<<dp[3][2]<<" ";
// cout<<dp[2][1]<<"\n";
cout<<dp[n][k]<<"\n";
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);cout.tie(0);
    ll t;
    cin>>t;
    while(t--)
    {
        solve();
    }
    return 0;
}

```