01背包 完全背包 多重背包

01背包

P1048 [NOIP2005 普及组] 采药

```cpp
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
#define ROF(i,a,b) for(int i=(a);i>=(b);--i)
using namespace std;

int w[105],val[105],dp[1005];

int main(){
    int t,m,res=-1;
    cin>>t>>m;
    FOR(i,1,m)
        cin>>w[i]>>val[i];
    FOR(i,1,m){
        ROF(j,t,0){
            if(j>=w[i]) dp[j]=max(dp[j-w[i]]+val[i],dp[j]);
        }
    }
    cout<<dp[t];
    return 0;
}
```

(变种)P1802 5 倍经验日

```cpp
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
#define ROF(i,a,b) for(int i=(a);i>=(b);--i)
using namespace std;

int win[1100],lose[1100],use[1100],dp[1100];

int main(){
    int n,m;
    cin>>n>>m;
    FOR(i,1,n)
        cin>>lose[i]>>win[i]>>use[i];
    FOR(i,1,n){
        ROF(j,m,0){
            if(j>=use[i]) dp[j]=max(dp[j]+lose[i],dp[j-use[i]]+win[i]);
            else dp[j]+=lose[i];
        }
    }
```

```
    cout<<5ll*dp[m];
    return 0;
}
```

完全背包

P1616 疯狂的采药

```
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
#define int long long
using namespace std;

const int N=1e4+5,M=1e7+5;
int n,m,w[N],val[N],dp[M];

signed main(){
    cin>>m>>n;
    FOR(i,1,n)
        cin>>w[i]>>val[i];
    FOR(i,1,n){
        FOR(j,w[i],m){
            dp[j]=max(dp[j],dp[j-w[i]]+val[i]);
        }
    }
    cout<<dp[m];
    return 0;
}
```

多重背包 二进制优化

P1776 宝物筛选

```
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
#define ROF(i,a,b) for(int i=(a);i>=(b);--i)
using namespace std;

const int maxn=1e6+7;
int n,m,ans,cnt=1;
int dp[maxn],w[maxn],v[maxn];

int main(){
    int val,wet,ni;//价值，重量，有几个
    cin>>n>>m;//种类，最大载重
    FOR(i,1,n){
```

```
        scanf("%d%d%d",&val,&wet,&ni);
        for(int j=1;j<=ni;j<<=1){
            v[++cnt]=j*val,w[cnt]=j*wet;
            ni-=j;
        }
        if(ni) v[++cnt]=val*ni,w[cnt]=wet*ni;
    }

    FOR(i,1,cnt)
        ROF(j,m,w[i])
            dp[j]=max(dp[j],dp[j-w[i]]+v[i]);
    cout<<dp[m]<<endl;
    return 0;
}
```

记忆化搜索

01背包

[LuoguOJ P1048 [NOIP2005 普及组] 采药](https://www.luogu.com.cn/problem/P1048)

```
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;

int vol[1007],val[1007],save[1007][1007];
int V,N;//V 背包容积，N 物品数量

int dfs(int OBJ,int VOL){//第obj个物品，背包剩余vol，dfs本身表示背包当前价值val
    if(VOL<0) return 0;
    if(OBJ>N) return 0;
    if(save[OBJ][VOL]!=0) return save[OBJ][VOL];
    int psb1,psb2=0;
    psb1=dfs(OBJ+1,VOL);
    if(VOL-vol[OBJ]>=0) psb2=dfs(OBJ+1,VOL-vol[OBJ])+val[OBJ];
    return save[OBJ][VOL]=max(psb1,psb2);
}

int main(){
    cin>>V>>N;
    FOR(i,1,N)
        cin>>vol[i]>>val[i];
    int ANS=dfs(1,V);
    cout<<ANS<<endl;
    return 0;
}
```

(变形)[P1077 [NOIP2012 普及组] 摆花]

```cpp
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;

const int maxn=107;
#define mod 1000007;
int n,m,a[maxn],mem[maxn][maxn];

int dfs(int x,int k){
    if(k>m) return 0;
    if(k==m) return 1;
    if(x==n+1) return 0;
    if(mem[x][k]) return mem[x][k];
    int ans=0;
    FOR(i,0,a[x]) ans=(ans+dfs(x+1,k+i))%mod;
    return mem[x][k]=ans;
}

int main(){
    cin>>n>>m;
    FOR(i,1,n)
        cin>>a[i];
    cout<<dfs(1,0);
    return 0;
}
```

分组背包

LuoguOJ P1757 通天之分组背包

```cpp
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);i++)
using namespace std;

struct node{
    int vol,val;//体积，价值
}e[1001][1001];//第GRP组的第i件物品

int save[1001][1001];//记忆数组
int len[1001];//存储每组物品的个数（实际上就是m）
int n,m;//物品数量，背包容积
const int maxGRP=100;

int dfs(int GRP,int VOL){//第GRP组物品，剩余空间VOL
    if(save[GRP][VOL]!=0) return save[GRP][VOL];
    if(GRP>maxGRP) return 0;
    int res=dfs(GRP+1,VOL);

```

```
    for(int i=1;i<=len[GRP];i++){
        if(VOL<e[GRP][i].vol) continue;
        res=max(res,dfs(GRP+1,VOL-e[GRP][i].vol)+e[GRP][i].val);
    }
    return save[GRP][VOL]=res;
}
int main(){
    cin>>m>>n;
    int vol,val,grp;
    FOR(i,1,n){
        cin>>vol>>val>>grp;
        e[grp][++len[grp]]={vol,val};
    }
    cout<<dfs(1,m);
    return 0;
}
```

GxustOJ #382. 程序猿的工资

```
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;

struct node{
    int vol,val;//体积，价值
}e[101][101];//第GRP组的第i件物品

int save[101][101];//记忆数组
int len[101];//存储每组物品的个数（实际上就是m）
int n,m;//组数，每组的个数

int dfs(int GRP,int VOL){//第GRP组物品，剩余空间VOL
    if(save[GRP][VOL]!=0) return save[GRP][VOL];
    if(GRP>n) return 0;
    int res=dfs(GRP+1,VOL);

    for(int i=1;i<=len[GRP];i++){
        if(VOL<e[GRP][i].vol) continue;
        res=max(res,dfs(GRP+1,VOL-e[GRP][i].vol)+e[GRP][i].val);
    }
    return save[GRP][VOL]=res;
}
int main(){
    cin>>n>>m;
    int tmp;
    FOR(i,1,n){
        FOR(j,1,m){
            cin>>tmp;
            e[i][++len[i]]={j,tmp};
```

```
        }
    }
    cout<<dfs(1,m);
    return 0;
}
```

区间动态规划

P1880 [NOI1995] 石子合并

```
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
#define ROF(i,a,b) for(int i=(a);i>=(b);--i)
const int INF (1<<30);
const int inf (-1<<30);
using namespace std;

int dp1[207][207],dp2[207][207];
int v[207],presum[207];

int sum(int i,int j){
    return presum[j]-presum[i-1];
}

int main(){
    int n;
    cin>>n;
    FOR(i,1,n){
        cin>>v[i];
        v[i+n]=v[i];//环形，复制一倍到尾部
    }
    FOR(i,1,2*n)
        presum[i]=presum[i-1]+v[i];

    FOR(i,1,2*n)
        ROF(j,i-1,1){
            dp1[j][i]=INF,dp2[j][i]=inf;
            FOR(k,j,i-1){
                dp1[j][i]=min(dp1[j][i],dp1[j][k]+dp1[k+1][i]+sum(j,i));
                dp2[j][i]=max(dp2[j][i],dp2[j][k]+dp2[k+1][i]+sum(j,i));
            }
        }
    int min0=INF,max0=inf;
    FOR(i,1,n){
        min0=min(min0,dp1[i][i+n-1]);
        max0=max(max0,dp2[i][i+n-1]);//求各个区间的最大值
    }
    cout<<min0<<'\n'<<max0;
```

```
    return 0;
}
```

P1063 [NOIP2006 提高组] 能量项链

```
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
#define ROF(i,a,b) for(int i=(a);i>=(b);--i)
#define mem(a) memset(a,0,sizeof(a))
using namespace std;

int dp[207][207],v[207];

int main(){
    int n,ans=0;
    cin>>n;
    FOR(i,1,n){
        cin>>v[i];
        v[i+n]=v[i];//环形，复制一倍到尾部
    }
    FOR(i,1,2*n)//遍历终点
        ROF(j,i-1,1)//从后向前推终点之前的起点
            FOR(k,j,i-1)//分界点
            //ROF(k,i-1,j) 正序或倒序都可以
                dp[j][i]=max(dp[j][i],v[j]*v[k+1]*v[i+1]+dp[j][k]+dp[k+1][i]);
                //为什么3个相乘的数组下标(j,k+1,i+1)不相邻呢，
                //因为他们之间的已经被合并掉了，
                //左边是j->k，右边是k+1->i，这是合并后的能量。
    FOR(i,1,n)
        ans=max(ans,dp[i][i+n-1]);//求各个区间的最大值
    cout<<ans<<endl;
    return 0;
}
```

状态压缩动态规划

P1896 [SCOI2005]互不侵犯

参考

修改了代码风格，对预处理部分注释作了修正

```
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;

long long sta[2005],sit[2005],f[15][2005][105];
int n,k,cnt=0;

```

```cpp
void dfs(int x,int num,int cur){
    if(cur>=n){//当前状态处理完毕（国王位置越界了）
        sit[++cnt]=x;//新建一个状态
        sta[cnt]=num;
        return;
    }
    dfs(x,num,cur+1);//cur位置不放国王
    dfs(x+(1<<cur),num+1,cur+2);//cur位置放国王，与它相邻的位置不能再放国王
}

bool compatible(int j,int x){
    if(sit[j] & sit[x]) return false;
    if((sit[j]<<1) & sit[x]) return false;
    if(sit[j] & (sit[x]<<1)) return false;
    return true;
}

int main(){
    cin>>n>>k;
    dfs(0,0,0);//先预处理一行的所有合法状态
    FOR(j,1,cnt)
        f[1][j][sta[j]] = 1;
    FOR(i,2,n)
        FOR(j,1,cnt)
            FOR(x,1,cnt){
                if(!compatible(j,x)) continue;//排除不合法转移
                FOR(l,sta[j],k)
                    f[i][j][l]+=f[i-1][x][l-sta[j]];
            }
    long long ans=0;
    FOR(i,1,cnt)
        ans+=f[n][i][k];//累加答案
    cout<<ans;
    return 0;
}
```

最长子序列问题

最长不上升序列长度和最长上升序列长度

P1020 [NOIP1999 普及组] 导弹拦截

思路来源

需要注意的点：

`dp1[i]`的含义：最长不上升子序列长度为`i`时，最优的结尾元素，而不是最长不上升子序列。

`AC` 代码

```cpp
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;

const int maxn=1e5+7;
int a[maxn],dp1[maxn],dp2[maxn];

int main() {
    int n=1;
    while(cin>>a[n]) ++n;
    n--;

    int len1=1,len2=1;
    dp1[1]=a[1];
    dp2[1]=a[1];

    FOR(i,2,n){
        if(dp1[len1]>=a[i]) dp1[++len1]=a[i];
        else{
            int p1=upper_bound(dp1+1,dp1+1+len1,a[i],greater<int>())-dp1;
            dp1[p1]=a[i];
        }
        if(dp2[len2]<a[i]) dp2[++len2]=a[i];
        else{
            int p2=lower_bound(dp2+1,dp2+1+len2,a[i],less<int>())-dp2;
            dp2[p2]=a[i];
        }
    }
    cout<<len1<<endl<<len2;
    return 0;
}

```

用记忆化搜索做的O(n^2)方法：

```cpp
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
#define ROF(i,a,b) for(int i=(a);i>=(b);--i)
using namespace std;

const int maxn=1e5+7;
int n,a[maxn];
int mem1[maxn],mem2[maxn];

int dfs1(int p){
    if(mem1[p]!=0) return mem1[p];
    int res=1;
    FOR(i,p+1,n)
        if(a[p]>=a[i]) res=max(res,dfs1(i)+1);
```

```
    return mem1[p]=res;
}

int dfs2(int p){
    if(mem2[p]!=0) return mem2[p];
    int res=1;
    FOR(i,p+1,n)
        if(a[p]<a[i]) res=max(res,dfs2(i)+1);
    return mem2[p]=res;
}

int main(){
    n=1;
    while(cin>>a[n]) ++n;
    n--;

    ROF(i,n,1){
        dfs1(i);
        dfs2(i);
    }
    int ans1=0,ans2=0;
    FOR(i,1,n){
        ans1=max(ans1,mem1[i]);
        ans2=max(ans2,mem2[i]);
    }
    cout<<ans1<<"\n"<<ans2;;
}
```

最长公共子序列长度

P1439 【模板】最长公共子序列

转化成最长不下降子序列

```
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
#define mem(a) memset(a,0,sizeof(a))
using namespace std;

#define map _map
const int maxn=1e5+7;
int a2[maxn],dp[maxn],map[maxn];

int main() {
    int n;cin>>n;
    int x;
    FOR(i,1,n){
        cin>>x;//x=a1[i]
        map[x]=i;//a1[i] -> i
    }
```

```
    FOR(i,1,n){
        cin>>x;
        a2[i]=map[x];
    }

    int len=1;
    dp[1]=a2[1];

    FOR(i,2,n){
        if(dp[len]<=a2[i]) dp[++len]=a2[i];
        else{
            int p1=upper_bound(dp+1,dp+1+len,a2[i],less<int>())-dp;
            dp[p1]=a2[i];
        }
    }
    cout<<len;
    return 0;
}
```