AtCoder Grand Contest 054

比赛链接

B - Greedy Division

题意

给出 n 个物品，第 i 个物品权重为 w_i。现有甲乙两人，要求将物品重新排列后依次分配每个物品。

分配物品满足如下规则：如果甲当前所得的物品权重和不大于乙，则将物品分给甲，否则分给乙。

问使得甲乙两人最后得所得物品权重和相等的排列方案共有多少种。

解法

假定甲得到物品的顺序依次为 $x_1,x_2\cdots x_k$，乙得到物品的顺序依次为 $y_1,y_2\cdots y_{n-k}$。

已知当序列 x,y 固定时物品的总排列也是固定的。考虑背包求出不考虑 $x_1,x_2\cdots x_k$ 顺序时甲所得物品权重为 $\frac {\sum w_i}2$ 的方案数 g_k。

则答案为 $\sum_{k=1}^nk!(n-k)!g_k$。设 $\text{dp}(i,j,k)$ 表示仅考虑前 i 个物品，甲所得物品总和为 j，已经选中 k 个物品的方案数，不难得到状态转移。

时间复杂度 $O(n^2\sum w_i)$。

```
const int MAXN=105,MAXV=1e4+5,mod=998244353;
int w[MAXN],f[MAXN],dp[2][MAXV][MAXN];
int main()
{
    int n=read_int();
    _for(i,0,n)w[i]=read_int();
    int pos=0;
    dp[pos][0][0]=1;
    _for(i,0,n){
        pos^=1;
        _for(j,0,MAXV)_rep(k,0,n)
        dp[pos][j][k]=dp[!pos][j][k];
        _for(j,w[i],MAXV)_rep(k,1,n)
        dp[pos][j][k]=(dp[pos][j][k]+dp[!pos][j-w[i]][k-1])%mod;
    }
    f[0]=1;
    _rep(i,1,n)f[i]=1LL*f[i-1]*i%mod;
    int s=0;
    _for(i,0,n)
    s+=w[i];
    if(s%2==1)
```

```
    puts("0");
    else{
        s/=2;
        int p=1,ans=0;
        _rep(i,1,n)
        ans=(ans+1LL*dp[pos][s][i]*f[i]%mod*f[n-i])%mod;
        enter(ans);
    }
    return 0;
}
```