动态规划练习

P1095 守望者的逃离(简单递推1)

题目链接:

[https://www.luogu.com.cn/problem/P1095]:

参考代码：(简单递推)

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
int dp[300005];
int main()
{
    int m, s, t;
    cin >> m >> s >> t;
    for (int i = 1; i <= t; i++)//闪烁的转移
    {
        if (m >= 10)
            m -= 10, dp[i] = dp[i - 1] + 60;
        else
            m += 4, dp[i] = dp[i - 1];
    }
    int x = 0;
    for (int i = 1; i <= t; i++)//走路的转移
    {
        dp[i] = max(dp[i], dp[i - 1] + 17);
        if (dp[i] >= s)//结束
        {
            x = i;
            break;
        }
    }
    if (x != 0)//输出
        cout << "Yes\n"<<x;
    else
        cout << "No\n"<<dp[t];



    return 0;
}
```

P1586 四方定理(简单递推2)

题目链接：

[https://www.luogu.com.cn/problem/P1586]:

查看代码：

```
#include<bits/stdc++.h>
#define ll long long
using namespace std;
template<class T>
void read(T& x)
{
    T res = 0, f = 1; char c = getchar();
    while (!isdigit(c)) {
        if (c == '-')f = -1; c = getchar();
    }
    while (isdigit(c)) {
        res = (res << 3) + (res << 1) + c - '0'; c = getchar();
    }
    x = res * f;
}
const ll N = 200000 + 10;
int num[N],cnt,n, t,mx= 32768,dp[N][10];
int main()
{
    dp[0][0] = 1;
    for (int i = 1; i * i <= mx; i++)
    {
        for (int j = i * i; j <= mx; j++)
        {
            for (int ti = 1; ti <= 4; ti++)
            {
                dp[j][ti] += dp[j - i * i][ti - 1];
            }
        }
    }
    read(t);
    while (t--)
    {
        read(n);

        ll ans = 0;
        for (int i = 1; i <= 4; i++)ans += dp[n][i];
        printf("%lld\n", ans);
    }
    return 0;
}

```

P3842 [TJOI2007]线段(简单递推3)

题目链接：

[https://www.luogu.com.cn/problem/P3842]:

参考代码：(无视快读，就当成scanf即可)

```
#include<bits/stdc++.h>

#define ll long long
using namespace std;
template<class T>
void read(T& x)
{
    T res = 0, f = 1; char c = getchar();
    while (!isdigit(c)) {
        if (c == '-')f = -1; c = getchar();
    }
    while (isdigit(c)) {
        res = (res << 3) + (res << 1) + c - '0'; c = getchar();
    }
    x = res * f;
}
const ll N = 200000 + 10;
int n, l[N], r[N];
int dpl[N], dpr[N];//到第i条线段左or右边的最短横向移动距离
int main()
{
    read(n);
    for (int i = 1; i <= n; i++)
    {
        read(l[i]), read(r[i]);
    }
    l[0] = 1, r[0] = 1;
    l[n+1] = n, r[n+1] = n;
    //0->n+1
    for (int i = 1; i <= n+1; i++)
    {
        int len = (r[i] - l[i]);//线段长度
        dpl[i] = min(dpl[i - 1]+abs(r[i]-l[i-1]), dpr[i - 1]+ abs(r[i] - r[i - 1])) +len ;//对应递推
        dpr[i] = min(dpl[i - 1]+ abs(l[i] - l[i - 1]), dpr[i - 1]+ abs(l[i] - r[i - 1])) +len ;//对应递推
    }
    printf("%d\n", min(dpl[n+1], dpr[n+1]) + n - 1);
    return 0;
}

```

P1091 合唱队形(两次最长升)

题目链接：

[https://www.luogu.com.cn/problem/P1091]:

参考代码：(两次最长升即可)

```cpp
#include<bits/stdc++.h>
using namespace std;
int n, a[105],s[105],f[105],ans;
int max(int a, int b)
{
    return a > b ? a : b;
}
int main()
{
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        cin >> a[i];
        s[i] = 1;
        f[i] = 1;
    }
    for (int i = 1; i <= n; i++)
        for (int j = 1; j < i; j++)
            if (a[i] > a[j])
                s[i] = max(s[i], s[j] + 1);//从前向后的转移，s[i]表示以i为结尾的最长的上升长度，从前向后
    for (int i = n; i >= 1; i--)
        for (int j = n; j > i; j--)
            if (a[i] > a[j])
                f[i] = max(f[i], f[j] + 1);//从后向前的转移，f[i]表示以i为结尾的最长的上升长度，从后向前
    for (int i = 1; i <= n; i++)
    {
        ans = max(ans, f[i] + s[i] - 1);//扫描一遍，取个max
    }
    cout <<n-ans;
    return 0;
}

```

P1387 最大正方形

题目链接：

[https://www.luogu.com.cn/problem/P1387]:

参考代码：

```cpp
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
int n,m,a[105][105],f[110][105],ans;
int main()
{
    cin >> n >> m;
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= m; j++)
            cin >> a[i][j];
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= m; j++)
        {
            if (a[i][j] == 1)
            {
                f[i][j] = min(min(f[i - 1][j], f[i][j - 1]), f[i - 1][j - 1]) + 1;
                ans = max(f[i][j], ans);
            }
        }
    cout << ans;
    return 0;
}
```

P1048 采药(简单背包1)

题目链接:

[https://www.luogu.com.cn/problem/P1048]:

参考代码:(简单背包问题)

空间未优化:

```
#include<bits/stdc++.h>

using namespace std;
#define N 2000
int t, m;
int v[N], w[N],dp[N][N];//dp[i][j]表示前i个草药，j个时间可以获取的最大价值
int main()
{
    scanf("%d%d", &t, &m);
    for (int i = 1; i <= m; i++)scanf("%d%d", &v[i], &w[i]);
    for(int i=1;i<=m;i++)
        for (int j = 1; j <= t; j++)
        {
            dp[i][j] = dp[i - 1][j];
            if (j >= v[i])dp[i][j] = max(dp[i][j], dp[i - 1][j - v[i]] + w[i]);
        }
    printf("%d\n", dp[m][t]);
    return 0;
}

```

空间优化:

```
#include<bits/stdc++.h>
using namespace std;
int v[105],w[105],dp[1005], sumv, n;;
int main()
{
    cin >> sumv >> n;
    for (int i = 1; i <= n; i++)
    {
        cin >> v[i] >> w[i];
    }
    for(int i=1;i<=n;i++)
        for (int j = sumv; j >= v[i]; j--)
        {
            dp[j] = max(dp[j], dp[j - v[i]] + w[i]);
        }
    cout << dp[sumv];
    return 0;
}
```

P1049 装箱问题(简单背包1)

题目链接:

[https://www.luogu.com.cn/problem/P1049]:

参考代码:(请无视快读，当成scanf)简单背包问题

```
#include<bits/stdc++.h>
#define ll long long
using namespace std;
template<class T>
void read(T& x)
{
    T res = 0, f = 1; char c = getchar();
    while (!isdigit(c)) {
        if (c == '-')f = -1; c = getchar();
    }
    while (isdigit(c)) {
        res = (res << 3) + (res << 1) + c - '0'; c = getchar();
    }
    x = res * f;
}
const ll N = 200000 + 10;
int v, n, w[N],dp[N];
int main()
{
    read(v), read(n);
    for (int i = 1; i <= n; i++)read(w[i]);
    for (int i = 1; i <= n; i++)
    {
        for (int j = v; j >= w[i]; j--)
        {
            dp[j] = max(dp[j], dp[j - w[i]] + w[i]);
        }
    }
    for (int i = 1; i <= v; i++)dp[i] = max(dp[i - 1], dp[i]);
    printf("%d\n", v - dp[v]);
    return 0;
}
```

P1064 金明的预算方案(有依赖的背包)

题目链接：

[https://www.luogu.com.cn/problem/P1064]

参考代码:(有依赖的背包问题)

```
#include<iostream>
#include<bits/stdc++.h>

using namespace std;
int m, n;
int mw[65], fw[65][3], mv[65], fv[65][3],f[50000];
int mx(int x, int y)
{
    return x > y ? x : y;
}
int main()
{
    cin >> n >> m;
    for (int i = 1; i <= m; i++)
    {
        int v, p, q;
        cin >> v >> p >> q;
        if (!q)
        {
            mv[i] = v;
            mw[i] = mv[i] * p;
        }
        else
        {
            fw[q][0]++;
            fv[q][fw[q][0]] = v;
            fw[q][fw[q][0]] = v * p;
        }
    }
    for(int i=1;i<=m;i++)
        for (int j = n; j >= mv[i]; j--)
        {
            f[j] = mx(f[j], f[j - mv[i]] + mw[i]);
            if (fw[i][0])
            {
                if (j >= mv[i] + fv[i][1])
                    f[j] = mx(f[j], f[j - mv[i] - fv[i][1]] + mw[i] + fw[i][1]);
                if (j >= mv[i] + fv[i][2])
                    f[j] = mx(f[j], f[j - mv[i] - fv[i][2]] + mw[i] + fw[i][2]);
                if (j >= mv[i] + fv[i][1]+fv[i][2])
                    f[j] = mx(f[j], f[j - mv[i] - fv[i][1] - fv[i][2]] + mw[i] + fw[i][1] + fw[i][2]);
            }
        }
    cout << f[n];
    return 0;
}
```

P1833 樱花(二进制优化的背包)

题目链接:

[https://www.luogu.com.cn/problem/P1833]:

参考代码：(无视快读)

```
#include<bits/stdc++.h>

#define ll long long
using namespace std;
template<class T>
void read(T& x)
{
    T res = 0, f = 1; char c = getchar();
    while (!isdigit(c)) {
        if (c == '-')f = -1; c = getchar();
    }
    while (isdigit(c)) {
        res = (res << 3) + (res << 1) + c - '0'; c = getchar();
    }
    x = res * f;
}
const ll N = 200000 + 10;
int sumv,h1,m1,h2,m2,n;
int t[N], c[N], p[N];
int st[N], sw[N],tot;
int dp[N];
int main()
{
    scanf("%d:%d%d:%d%d", &h1, &m1, &h2, &m2,&n);
    if (m2 < m1)
        m2 += 60, h2 -=1;
    sumv = (h2 - h1) * 60 + m2 - m1;//计算总时间差
    for (int i = 1; i <= n; i++)
    {
        scanf("%d%d%d", &t[i], &c[i], &p[i]);
        if (p[i] == 0)
        {
            p[i] = 999999;
        }
    }
    for (int i = 1; i <= n; i++)
    {
        for (int j = 0; j <= 30; j++)
        {
            int w = (1 << j);//表示要看w次
            if (p[i] > w)
            {
                tot++;
                st[tot] = t[i] * w;
                sw[tot] = c[i] * w;
                p[i] -= w;//看过w次了
            }
            else
            {
                tot++;
                st[tot] = t[i] * p[i];
                sw[tot] = c[i] * p[i];//全看完
                p[i] = 0;
                break;
            }
        }
    }
    for (int i = 1; i <= tot; i++)
    {
        for (int j = sumv; j >= st[i]; j--)
        {
            dp[j] = max(dp[j - st[i]] + sw[i], dp[j]);
        }
    }
    for (int i = 1; i <= sumv; i++)dp[i] = max(dp[i], dp[i - 1]);
    printf("%d\n", dp[sumv]);
    return 0;
}

```

P1020 导弹拦截(没自闭的话能讲下，最长升nlogn的解法)

题目链接

[https://www.luogu.com.cn/problem/P1020]:

参考代码：nlogn求最长升的方法(其实线段树暴力转移即可)

```
#include<bits/stdc++.h>
#define ll long long
using namespace std;
const ll N=1000000+10;
int a[N],ans1,ans2,n,dps[N],dpx[N];

int main()
{
    while(scanf("%d",&a[++n])!=EOF&&a[n]);
    n--;
    for(int i=1;i<=n;i++)
    {
        if(i==1)
        {
            dps[++ans1]=a[1],dpx[++ans2]=a[1];
            continue;
        }
        if(a[i]<=dpx[ans2])
            dpx[++ans2]=a[i];
        else
            dpx[upper_bound(dpx+1,dpx+1+ans2,a[i],greater<int>())-(dpx)]=a[i];
        if(a[i]>dps[ans1])
            dps[++ans1]=a[i];
        else
            dps[lower_bound(dps+1,dps+1+ans1,a[i])-(dps)]=a[i];
    }
    printf("%d\n%d",ans2,ans1);
    return 0;
}

```