

## 动态规划练习

### 动态规划练习

P1095 守望者的逃离(简单递推1)  
P1586 四方定理(简单递推2)  
P3842 [TJOI2007]线段(简单递推3)  
P1091 合唱队形(两次最长升)  
P1387 最大正方形  
P1048 采药(简单背包1)  
P1049 装箱问题(简单背包1)  
P1064 金明的预算方案(有依赖的背包)  
P1833 樱花(二进制优化的背包)  
P1020 导弹拦截(没自闭的话能讲下, 最长升 $n\log n$ 的解法)

## 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)

```
1  #include<iostream>
2  #include<bits/stdc++.h>
3  using namespace std;
4  int dp[300005];
5  int main()
6  {
7      int m, s, t;
8      cin >> m >> s >> t;
9      for (int i = 1; i <= t; i++)//闪烁的转移
10     {
11         if (m >= 10)
12             m -= 10, dp[i] = dp[i - 1] + 60;
13         else
14             m += 4, dp[i] = dp[i - 1];
15     }
16     int x = 0;
17     for (int i = 1; i <= t; i++)//走路的转移
18     {
19         dp[i] = max(dp[i], dp[i - 1] + 17);
20         if (dp[i] >= s)//结束
21         {
22             x = i;
23             break;
24         }
25     }
26     if (x != 0)//输出
27         cout << "Yes\n"<<x;
28     else
29         cout << "No\n"<<dp[t];
30
31 }
```

```
32  
33     return 0;  
34 }
```

## P1586 四方定理(简单递推2)

题目链接:

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

查看代码:

```
1  #include<bits/stdc++.h>  
2  #define ll long long  
3  using namespace std;  
4  template<class T>  
5  void read(T& x)  
6  {  
7      T res = 0, f = 1; char c = getchar();  
8      while (!isdigit(c)) {  
9          if (c == '-')f = -1; c = getchar();  
10     }  
11     while (isdigit(c)) {  
12         res = (res << 3) + (res << 1) + c - '0'; c = getchar();  
13     }  
14     x = res * f;  
15 }  
16 const ll N = 200000 + 10;  
17 int num[N], cnt, n, t, mx = 32768, dp[N][10];  
18 int main()  
19 {  
20     dp[0][0] = 1;  
21     for (int i = 1; i * i <= mx; i++)  
22     {  
23         for (int j = i * i; j <= mx; j++)  
24         {  
25             for (int ti = 1; ti <= 4; ti++)  
26             {  
27                 dp[j][ti] += dp[j - i * i][ti - 1];  
28             }  
29         }  
30     }  
31     read(t);  
32     while (t--)  
33     {  
34         read(n);  
35  
36         ll ans = 0;  
37         for (int i = 1; i <= 4; i++) ans += dp[n][i];  
38         printf("%lld\n", ans);  
39     }  
40     return 0;  
41 }  
42
```

## 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)

```
1  #include<bits/stdc++.h>
2
3  #define ll long long
4  using namespace std;
5  template<class T>
6  void read(T& x)
7  {
8      T res = 0, f = 1; char c = getchar();
9      while (!isdigit(c)) {
10         if (c == '-')f = -1; c = getchar();
11     }
12     while (isdigit(c)) {
13         res = (res << 3) + (res << 1) + c - '0'; c = getchar();
14     }
15     x = res * f;
16 }
17 const ll N = 200000 + 10;
18 int n, l[N], r[N];
19 int dpl[N], dpr[N]; //到第i条线段左or右边的最短横向移动距离
20 int main()
21 {
22     read(n);
23     for (int i = 1; i <= n; i++)
24     {
25         read(l[i]), read(r[i]);
26     }
27     l[0] = 1, r[0] = 1;
28     l[n+1] = n, r[n+1] = n;
29     //0->n+1
30     for (int i = 1; i <= n+1; i++)
31     {
32         int len = (r[i] - l[i]); //线段长度
33         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
34         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
35     }
36     printf("%d\n", min(dpl[n+1], dpr[n+1]) + n - 1);
37     return 0;
38 }
39
```

## P1091 合唱队形(两次最长升)

题目链接:

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

参考代码: (两次最长升即可)

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int n, a[105], s[105], f[105], ans;
4  int max(int a, int b)
5  {
6      return a > b ? a : b;
7  }
8  int main()
9  {
10     cin >> n;
11     for (int i = 1; i <= n; i++)
12     {
13         cin >> a[i];
14         s[i] = 1;
15         f[i] = 1;
16     }
17     for (int i = 1; i <= n; i++)
18         for (int j = 1; j < i; j++)
19             if (a[i] > a[j])
20                 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
21     for (int i = n; i >= 1; i--)
22         for (int j = n; j > i; j--)
23             if (a[i] > a[j])
24                 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
25     for (int i = 1; i <= n; i++)
26     {
27         ans = max(ans, f[i] + s[i] - 1); //扫描一遍, 取个max
28     }
29     cout << n - ans;
30     return 0;
31 }
32
```

## P1387 最大正方形

题目链接:

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

参考代码:

```
1  #include<iostream>
2  #include<bits/stdc++.h>
3  using namespace std;
4  int n, m, a[105][105], f[110][105], ans;
5  int main()
6  {
```

```

7      cin >> n >> m;
8      for (int i = 1; i <= n; i++)
9          for (int j = 1; j <= m; j++)
10             cin >> a[i][j];
11      for (int i = 1; i <= n; i++)
12          for (int j = 1; j <= m; j++)
13              {
14                  if (a[i][j] == 1)
15                      {
16                          f[i][j] = min(min(f[i - 1][j], f[i][j - 1]), f[i - 1][j -
17                          1]) + 1;
18                          ans = max(f[i][j], ans);
19                      }
20              }
21      cout << ans;
22      return 0;
23  }

```

## 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:

```

1  #include<bits/stdc++.h>
2
3  using namespace std;
4  #define N 2000
5  int t, m;
6  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
7  int main()
8  {
9      scanf("%d%d", &t, &m);
10     for (int i = 1; i <= m; i++) scanf("%d%d", &v[i], &w[i]);
11     for (int i = 1; i <= m; i++)
12         for (int j = 1; j <= t; j++)
13             {
14                 dp[i][j] = dp[i - 1][j];
15                 if (j >= v[i]) dp[i][j] = max(dp[i][j], dp[i - 1][j - v[i]] +
16                 w[i]);
17             }
18     printf("%d\n", dp[m][t]);
19     return 0;
20 }

```

空间优化:

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int v[105], w[105], dp[1005], sumv, n;;

```

```

4  int main()
5  {
6      cin >> sumv >> n;
7      for (int i = 1; i <= n; i++)
8      {
9          cin >> v[i] >> w[i];
10     }
11     for(int i=1;i<=n;i++)
12         for (int j = sumv; j >= v[i]; j--)
13         {
14             dp[j] = max(dp[j], dp[j - v[i]] + w[i]);
15         }
16     cout << dp[sumv];
17     return 0;
18 }

```

## P1049 装箱问题(简单背包1)

题目链接:

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

```

1  #include<bits/stdc++.h>
2  #define ll long long
3  using namespace std;
4  template<class T>
5  void read(T& x)
6  {
7      T res = 0, f = 1; char c = getchar();
8      while (!isdigit(c)) {
9          if (c == '-')f = -1; c = getchar();
10     }
11     while (isdigit(c)) {
12         res = (res << 3) + (res << 1) + c - '0'; c = getchar();
13     }
14     x = res * f;
15 }
16 const ll N = 200000 + 10;
17 int v, n, w[N], dp[N];
18 int main()
19 {
20     read(v), read(n);
21     for (int i = 1; i <= n; i++)read(w[i]);
22     for (int i = 1; i <= n; i++)
23     {
24         for (int j = v; j >= w[i]; j--)
25         {
26             dp[j] = max(dp[j], dp[j - w[i]] + w[i]);
27         }
28     }
29     for (int i = 1; i <= v; i++)dp[i] = max(dp[i - 1], dp[i]);
30     printf("%d\n", v - dp[v]);
31     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)

```

1  #include<iostream>
2  #include<bits/stdc++.h>
3
4  using namespace std;
5  int m, n;
6  int mw[65], fw[65][3], mv[65], fv[65][3], f[50000];
7  int mx(int x, int y)
8  {
9      return x > y ? x : y;
10 }
11 int main()
12 {
13     cin >> n >> m;
14     for (int i = 1; i <= m; i++)
15     {
16         int v, p, q;
17         cin >> v >> p >> q;
18         if (!q)
19         {
20             mv[i] = v;
21             mw[i] = mv[i] * p;
22         }
23         else
24         {
25             fw[q][0]++;
26             fv[q][fw[q][0]] = v;
27             fw[q][fw[q][0]] = v * p;
28         }
29     }
30     for(int i=1;i<=m;i++)
31         for (int j = n; j >= mv[i]; j--)
32         {
33             f[j] = mx(f[j], f[j - mv[i]] + mw[i]);
34             if (fw[i][0])
35             {
36                 if (j >= mv[i] + fv[i][1])
37                     f[j] = mx(f[j], f[j - mv[i] - fv[i][1]] + mw[i] + fw[i]
[1]);
38                 if (j >= mv[i] + fv[i][2])
39                     f[j] = mx(f[j], f[j - mv[i] - fv[i][2]] + mw[i] + fw[i]
[2]);
40                 if (j >= mv[i] + fv[i][1]+fv[i][2])
41                     f[j] = mx(f[j], f[j - mv[i] - fv[i][1] - fv[i][2]] +
mw[i] + fw[i][1] + fw[i][2]);
42             }

```

```

43     }
44     cout << f[n];
45     return 0;
46 }

```

## 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)

```

1  #include<bits/stdc++.h>
2
3  #define ll long long
4  using namespace std;
5  template<class T>
6  void read(T& x)
7  {
8      T res = 0, f = 1; char c = getchar();
9      while (!isdigit(c)) {
10         if (c == '-')f = -1; c = getchar();
11     }
12     while (isdigit(c)) {
13         res = (res << 3) + (res << 1) + c - '0'; c = getchar();
14     }
15     x = res * f;
16 }
17 const ll N = 200000 + 10;
18 int sumv, h1, m1, h2, m2, n;
19 int t[N], c[N], p[N];
20 int st[N], sw[N], tot;
21 int dp[N];
22 int main()
23 {
24     scanf("%d:%d%d:%d", &h1, &m1, &h2, &m2, &n);
25     if (m2 < m1)
26         m2 += 60, h2 -= 1;
27     sumv = (h2 - h1) * 60 + m2 - m1; // 计算总时间差
28     for (int i = 1; i <= n; i++)
29     {
30         scanf("%d%d", &t[i], &c[i], &p[i]);
31         if (p[i] == 0)
32         {
33             p[i] = 999999;
34         }
35     }
36     for (int i = 1; i <= n; i++)
37     {
38         for (int j = 0; j <= 30; j++)
39         {
40             int w = (1 << j); // 表示要看w次
41             if (p[i] > w)
42                 {

```



```

43         tot++;
44         st[tot] = t[i] * w;
45         sw[tot] = c[i] * w;
46         p[i] -= w; //看过w次了
47     }
48     else
49     {
50         tot++;
51         st[tot] = t[i] * p[i];
52         sw[tot] = c[i] * p[i]; //全看完
53         p[i] = 0;
54         break;
55     }
56 }
57 }
58 for (int i = 1; i <= tot; i++)
59 {
60     for (int j = sumv; j >= st[i]; j--)
61     {
62         dp[j] = max(dp[j - st[i]] + sw[i], dp[j]);
63     }
64 }
65 for (int i = 1; i <= sumv; i++) dp[i] = max(dp[i], dp[i - 1]);
66 printf("%d\n", dp[sumv]);
67 return 0;
68 }
69

```

## P1020 导弹拦截(没自闭的话能讲下，最长升nlogn的解法)

### 题目链接

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

```

1  #include<bits/stdc++.h>
2  #define ll long long
3  using namespace std;
4  const ll N=1000000+10;
5  int a[N],ans1,ans2,n,dps[N],dpx[N];
6
7  int main()
8  {
9      while(scanf("%d",&a[++n])!=EOF&&a[n]);
10     n--;
11     for(int i=1;i<=n;i++)
12     {
13         if(i==1)
14         {
15             dps[++ans1]=a[1],dpx[++ans2]=a[1];
16             continue;
17         }
18         if(a[i]<=dpx[ans2])
19             dpx[++ans2]=a[i];
20     }
21 }

```

```
20         else
21             dpx[upper_bound(dpx+1,dpx+1+ans2,a[i],greater<int>())-
(dpx)]=a[i];
22             if(a[i]>dps[ans1])
23                 dps[++ans1]=a[i];
24             else
25                 dps[lower_bound(dps+1,dps+1+ans1,a[i])-(dps)]=a[i];
26     }
27     printf("%d\n%d",ans2,ans1);
28     return 0;
29 }
30
```