

DP

dp单调序列

```
1  /*
2  1·b(I,1)表示第I个数的数值本身;
3  2·b(I,2)表示从I位置到达N的最长不下降序列长度
4  3·b(I,3)表示从I位置开始最长不下降序列的下一个位置
5  */
6  #include<iostream>
7  #include<cstdio>
8  using namespace std;
9  int n,i,j,l,k,b[200][10];
10 int main()
11 {
12     scanf("%d",&n);
13     for (i=1;i<=n;i++){
14         cin>>b[i][1];
15         b[i][2]=1;b[i][3]=0;
16     }
17     for (i=n-1;i>=1;i--){
18         l=0;k=0;
19         for (j=i+1;j<=n;j++){
20             if ((b[j][1]>b[i][1])&&(b[j][2]>l)){
21                 l=b[j][2];
22                 k=j;
23             }
24             if (l>0){
25                 b[i][2]=l+1;b[i][3]=k;
26             }
27         }
28         k=1;
29         for (j=1;j<=n;j++){
30             if (b[j][2]>b[k][2]) k=j;
31             cout<<"max="<<b[k][2]<<endl;
32             while (k!=0){
33                 cout<<' '<<b[k][1];
34                 k=b[k][3];
35             }
36             return 0;
37         }
38     }
```

01背包

```
1  #include<iostream>
2  #include<cstdio>
3  using namespace std;
4  int m,n;
5  int w[352000],val[352000],f[352000];
```

```

6  int main(){
7      scanf("%d%d",&n,&m);
8      int x,y;
9      for(int i=1;i<=n;i++){
10         scanf("%d%d",&w[i],&val[i]);
11     }
12     for(int i=1;i<=n;i++){
13         for(int j=m;j>=w[i];j--){
14             f[j]=max(f[j],f[j-w[i]]+val[i]);
15         }
16     }
17     printf("%d\n",f[m]);
18     return 0;
19 }

```

完全背包

```

1  #include<iostream>
2  #include<cstring>
3  #include<cstdio>
4  using namespace std;
5  //完全背包
6  int n,V;
7  int f[100020];
8  int v[20000],w[20000];
9  int main(){
10     scanf("%d%d",&n,&V);
11     for(int i=1;i<=n;i++){
12         scanf("%d",&v[i]);
13         scanf("%d",&w[i]);
14     }
15     for(int i=1;i<=n;i++){
16         for(int j=v[i];j<=V;j++){
17             f[j]=max(f[j],f[j-v[i]]+w[i]);
18         }
19     }
20     int maxn=0;
21     for(int i=1;i<=n;i++){
22         maxn=max(maxn,f[i]);
23     }
24     printf("%d",maxn);
25
26     return 0;
27 }

```