

ICPC CodeBook

1 Dynamic Programming

1.1 0/1 Knapsack_problems

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int f[1000]={0};
4 int n=0, m=0;
5 int main(){
6     cin >> n >> m;
7     for (int i = 1; i <= n; i++){
8         int price = 0, value = 0;
9         cin >> price >> value;
10        for (int j = m; j >= price; j--){
11            if (f[j-price]+value>f[j]){
12                f[j]=f[j-price]+value;
13            }
14        }
15    }
16    cout << f[m] << endl;
17    return 0;
18 }
```

1.2 Complete_Knapsack_problems

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int f[1000]={0};
4 int n=0, m=0;
5 int main(){
6     cin >> n >> m;
7     for (int i=1; i<=n; i++){
8         int price=0, value=0;
9         cin >> price >> value;
10        for (int j=price; j<=m; j++){
11            if (f[j-price]+value>f[j]){
12                f[j]=f[j-price]+value;
13            }
14        }
15    }
16    cout << f[m] << endl;
17    return 0;
18 }
```

1.3 LCS

```

1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int dp[1001][1001];
5 int lcs(const string &s, const string &t){
6     int m = s.size(), n = t.size();
7     if (m == 0 || n == 0){
8         return 0;
9     }
10    for(int i = 0; i <= m; ++i){
11        dp[i][0] = 0;
12    }
13    for(int j = 1; j <= n; ++j){
14        dp[0][j] = 0;
15    }
16    for(int i = 0; i < m; ++i){
17        for (int j = 0; j < n; ++j){
18            if(s[i] == t[j]){
19                dp[i+1][j+1] = dp[i][j]+1;
20            }else{
21                dp[i+1][j+1] = max(dp[i+1][j],
22                                dp[i][j+1]);
23            }
24        }
25    }
26 }
```

```

22     }
23     }
24 }
25     return dp[m][n];
26 }
```

1.4 LICS

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int a[100] = {0};
4 int b[100] = {0};
5 int f[100] = {0};
6 int n = 0, m = 0;
7 int main(){
8     cin >> n;
9     for(int i = 1; i <= n; i++){
10        cin >> a[i];
11    }
12    cin >> m;
13    for(int i = 1; i <= m; i++){
14        cin >> b[i];
15    }
16    for(int i = 1; i <= n; i++){
17        int k = 0;
18        for (int j = 1; j <= m; j++){
19            if(a[i] > b[j] && f[j] > k){
20                k = f[j];
21            }else if(a[i] == b[j] && k + 1 > f[j]){
22                f[j] = k + 1;
23            }
24        }
25    }
26    int ans=0;
27    for(int i = 1; i <= m; i++){
28        if(f[i] > ans){
29            ans = f[i];
30        }
31    }
32    cout << ans << endl;
33    return 0;
34 }
```

1.5 LIS

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int n=0;
4 int a[100]={0}, f[100]={0}, x[100]={0};
5 int main(){
6     cin >> n;
7     for(int i = 1; i <= n; i++){
8         cin >> a[i];
9         x[i] = INT_MAX;
10    }
11    f[0]=0;
12    int ans=0;
13    for(int i = 1; i <= n; i++){
14        int l = 0, r = i;
15        while (l+1<r){
16            int m=(l+r)/2;
17            if (x[m]<a[i]){
18                l=m;
19            }else{
20                r=m;
21            }
22            // change to x[m]<=a[i] for
            // non-decreasing case
23        }
24        f[i]=l+1;
25        x[l+1]=a[i];
26        if(f[i]>ans){
27            ans=f[i];
28        }
29    }
30 }
```

```
28     }  
29     }  
30     cout << ans << endl;  
31     return 0;  
32 }
```