ICPC CodeBook

1 Basic

1.1 .vimrc

```
1 syntax enable
2 set nu
3 set cursorline
4 set ts=2 sts=2 sw=2 et ai
5 set mouse=a
6 set wrap
7 set showcmd
8 set backspace=indent,eol,start
9
10 inoremap ( ()<ESC>i
11 inoremap [ []<ESC>i
12 inoremap {<CR> {<CR>}<ESC>ko
```

2 Dynamic Programming

2.1 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(){
       cin >> n >> m;
       for (int i = 1; i \le n; i++){
7
8
           int price = 0, value = 0;
9
           cin >> price >> value;
           for (int j = m; j >= price; j--){
10
11
                if (f[j-price]+value>f[j]){
12
                    f[j]=f[j-price]+value;
13
           }
14
15
16
       cout << f[m] << endl;</pre>
       return 0;
17
18 }
```

2.2 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++){</pre>
           int price=0, value=0;
8
9
           cin >> price >> value;
           for (int j=price; j<=m; j++){</pre>
10
                if (f[j-price]+value>f[j]){
11
12
                     f[j]=f[j-price]+value;
                }
13
14
           }
15
       cout << f[m] << endl;</pre>
16
17
       return 0;
18 }
```

2.3 LCS

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

2.4 LICS

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

2.5 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;
```

13

14

15

16

17 }

for(auto el : p){

int i = c[el];

pos[i] ++;

p_new[pos[i]] = el;

```
for(int i = 1; i <= n; i++){</pre>
            cin >> a[i];
8
9
            x[i] = INT_MAX;
       }
10
11
       f[0]=0;
       int ans=0;
12
13
       for(int i = 1; i \le n; i++){
14
            int 1 = 0,r = i;
            while (1+1<r){</pre>
15
                 int m=(1+r)/2;
16
                 if (x[m]<a[i]){
17
18
                      1=m;
19
                 }else{
20
                      r=m;
21
                 }
                 // change to x[m] \le a[i] for
22
                      non-decreasing case
23
            f[i]=1+1;
24
25
            x[l+1]=a[i];
            if(f[i]>ans){
26
27
                 ans=f[i];
28
29
30
       cout << ans << endl;</pre>
31
       return 0;
32 }
```

Data Structure

3.1 Disjoint Set Union-Find

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
  vector<int> dsu, rk;
6
  void initDSU(int n){
       dsu.resize(n);
8
       rk.resize(n);
9
       for(int i = 0; i < n; i++) dsu[i] = i, rk[i] = 1;</pre>
10 }
11
12 int findDSU(int x){
13
       if(dsu[x] == x) return x;
       dsu[x] = findDSU(dsu[x]);
14
15
       return dsu[x];
16 }
17
18 void unionDSU(int a, int b){
19
       int pa = findDSU(a), pb = findDSU(b);
20
       if(rk[pa] > rk[pb]) swap(pa, pb);
       if(rk[pa] == rk[pb]) rk[pb]++;
21
       dsu[pa] = pb;
22
23 }
```

String

4.1 Suffix Array

```
1 #include <bits/stdc++.h>
2 #define int long long
4 using namespace std;
6 void count_sort(auto &p, auto &c){
7
    int n = p.size();
    vector<int> cnt(n);
8
    for(auto el : c) cnt[el] ++;
    vector<int> p_new(n), pos(n);
10
    pos[0] = 0;
```

```
18
     p = p_new;
19 }
20
21 signed main(){
     string s;
22
23
     cin>>s;
24
     s += "$";
     int n = s.size();
25
26
     vector<pair<char, int>> v(n);
27
     vector<int> p(n), c(n);
     for(int i=0;i<n;i++) v[i] = {s[i], i};</pre>
28
29
     sort(v.begin(), v.end());
30
31
     for(int i=0;i<v.size();i++) p[i] = v[i].second;</pre>
32
     c[p[0]] = 0;
33
     for(int i=1;i<v.size();i++){</pre>
       if(v[i].first == v[i-1].first) c[p[i]] =
34
           c[p[i-1]];
35
       else c[p[i]] = c[p[i-1]] + 1;
36
37
     int k = 0;
38
     while ((1 << k) < n){
39
       for(int i=0; i< n; i++) p[i] = (p[i] - (1 << k) + n)
40
           % n;
41
       count_sort(p, c);
42
43
       vector < int > c_new(n);
44
       c_new[p[0]] = 0;
45
       for(int i=1;i<v.size();i++){</pre>
46
         pair<int, int> prev = {c[p[i-1]], c[(p[i-1] +
              (1 << k)) % n]};
47
         pair<int, int> now = {c[p[i]], c[(p[i] + (1 <<
              k)) % n]};
         if(prev == now) c_new[p[i]] = c_new[p[i-1]];
48
49
         else c_new[p[i]] = c_new[p[i-1]] + 1;
50
       }
       c = c_new;
51
52
       k++:
53
     for(int i=0;i<n;i++) cout<<p[i]<<"\n";</pre>
54
55 }
```

for(int i=1;i<n;i++) pos[i] = pos[i-1] + cnt[i-1];</pre>

4.2 Suffix Array LCP

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
5 vector<int> lcp(n);
6 \mid \mathbf{int} \mid k = 0;
7
  for(int i=0;i<n-1;i++){</pre>
8
       int pi = c[i];
9
       int j = p[pi - 1];
       while(s[i+k] == s[j+k]) k++;
10
11
       lcp[pi] = k;
12
       k = k-1 > 0 ? k-1 : 0;
13 }
```