

Programming 3-1: 动态规划

MA Jun

2018.09.10

1 Problem 3-1-A: "OJ-oriented" Problem Solving

1.1 AC Program

```
1 #include<cstdio>
2 #include<cstring>
3 #include<algorithm>
4 using namespace std;
5 const int N=1e4;
6 int T,n,a[N],dp[N];
7 int main(){
8     scanf("%d",&T);
9     while(T--){
10         scanf("%d",&n);
11         for (int i=1;i<=n;i++){
12             scanf("%d",a+i);
13             a[i]-=i;
14         }
15         dp[0]=0;
16         int cnt=0;
17         for(int i=1;i<=n;i++){
18             if (a[i]>=dp[cnt]){
19                 dp[++cnt]=a[i];
20             }
21             else {
22                 int tmp=upper_bound(dp+1,dp+cnt+1,a[i])-dp;
23                 dp[tmp]=a[i];
24             }
25         }
26         printf("%d\n",n-cnt);
27     }
28 }
29
30 /*****
31 Problem: 1535
32 User: 171860547
33 Language: C++
34 Result: J
35 Time:4 ms
36 Memory:1036 kb
37 *****/
```

Listing 1: A by 171860547

2 Problem 3-1-B:

2.1

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  const int INF = 0x3f3f3f3f;
4
5  int n = 0, hashing = 0;
6  bool s[18] = {};
7  int e[18][18] = {};
8  int ans[18][65540] = {};
9  int delivery[18][65540] = {};
10
11 int dp(int pos, int hashing){
12     if(ans[pos][hashing] > 0) {
13         return ans[pos][hashing];
14     }
15     if (hashing == 0) {
16         return 0;
17     }
18     int ret = INF, cur = INF;
19     for (int i = 1; i <= n; ++i) {
20         if ((hashing >> (i-1)) & 1) {
21             cur = dp(i, hashing - (1 << (i-1))) + e[pos][i];
22             if (cur <= ret) {
23                 ans[pos][hashing] = ret = cur;
24                 delivery[pos][hashing] = i;
25             }
26         }
27     }
28     return ret;
29 }
30
31 int main() {
32     int a = 0, b = 1, p = 0;
33
34     scanf("%d", &n);
35     for (int i = 0; i < n; ++i) {
36         scanf("%d", &a);
37         if (a) {
38             s[i] = true;
39             hashing += b;
40         }
41         b <<= 1;
42     }
43     for (int i = 0; i <= n; ++i) {
44         for (int j = 0; j <= n; ++j) {
45             scanf("%d", &e[i][j]);
46         }
47     }
48
49     printf("%d\n", dp(0, hashing));
50     while (hashing != 0) {
51         printf("%d ", delivery[p][hashing]);
52         p = delivery[p][hashing];

```

```
53     hashing -= (1 << (p-1));
54 }
55 return 0;
56 }
57 /*****
58     Problem: 1536
59     User: 171860508
60     Language: C++
61     Result:   J
62     Time:16 ms
63     Memory:10768 kb
64 *****/
```

Listing 2: B by 171860508

3 Problem 3-1-C: 2

3.1

```

1  #include<stdio.h>
2  #include<math.h>
3  int m, n;
4  char a[32][32];
5  long long int f[32][31]={0}; // int» 1»
6  int pos[31][2];
7  void gets(){
8      scanf("%d%d",&m,&n);
9      for(int i=1; i<=m; i++){
10         scanf("%s",&a[i]);
11         pos[i][0] = 0;
12         pos[i][1] = n;
13         for(int j=0; j<n; j++)
14             if(a[i][j]=='B')
15                 pos[i][0]=j+1;
16         for(int j=n-1; j>=0; j--)
17             if(a[i][j]=='R')
18                 pos[i][1]=j;
19     }
20     pos[m+1][0] = pos[m][0];
21     pos[m+1][1] = pos[m][1];
22 }
23
24 long long int dp(int i,int j){
25     if(pos[i][0]>j || pos[i][1]<j)
26         return 0;
27     if(f[i][j]!=0)
28         return f[i][j];
29     if(i==1)
30         return 1;
31     for(int k=j; k<=n; k++){
32         f[i][j]+=dp(i-1,k);
33     }
34     return f[i][j];
35 }
36 void puts(){
37     printf("%11d\n",f[m+1][pos[m][0]]);
38 }
39 int main(){
40     gets();
41     dp(m+1,pos[m][0]);
42     puts();
43 }
44 /*****
45 Problem: 1537
46 User: 171240511
47 Language: C++
48 Result: J
49 Time:0 ms
50 Memory:968 kb
51 *****/

```

Listing 3: C by 171240511