

SOFT500327: 算法设计与分析 2022-Fall

Homework-2 Solutions

软件 2101 杨豪 学号: 2206213297

2022 年 10 月 23 日

Honor Code: I promise that I finished the homework solutions on my own without copying other people's work.

pdf 中所示源码均附于 src/文件夹中

3-2

代码

```
3-2 > 3-2.cpp > DPmin(n)
1  #include <iostream>
2
3  using namespace std;
4  const int N = 110;
5  int dp[N][N], num[N][N];
6  // DP(i,j) is score from i to j.
7  // DPmin(i,j) = min{DP(i,k) + DP(k+1,j) + num[i][i+j]}
8  int DPmin(int n)
9  {
10     for (int j = 2; j < n; j++){
11         for (int i = 0; i + j < n; i++){
12             for (int k = 0; k < j; k++){
13                 if (k == 0){
14                     num[i][i+j] = num[i][i+k] + num[i+k+1][i+j];
15                     dp[i][i+j] = dp[i][i+k] + dp[i+k+1][i+j] + num[i][i+j];
16                 }
17                 else dp[i][i+j] = min(dp[i][i+j], dp[i][i+k] + dp[i+k+1][i+j] + num[i][i+j]);
18             }
19         }
20     }
21     return dp[0][n-1];
22 }
23
24 // DPmax(i,j) = max{DP(i,k) + DP(k+1,j) + num[i][i+j]}
25 int DPmax(int n)
26 {
27     for (int j = 2; j < n; j++){
28         for (int i = 0; i + j < n; i++){
29             for (int k = 0; k < j; k++){
30                 if (k == 0){
31                     num[i][i+j] = num[i][i+k] + num[i+k+1][i+j];
32                     dp[i][i+j] = dp[i][i+k] + dp[i+k+1][i+j] + num[i][i+j];
33                 }
34                 else dp[i][i+j] = max(dp[i][i+j], dp[i][i+k] + dp[i+k+1][i+j] + num[i][i+j]);
35             }
36         }
37     }
38     return dp[0][n-1];
39 }
40
41 int main()
42 {
43     freopen("input.txt", "r", stdin);
44     freopen("output.txt", "w", stdout);
45     int n;
46     cin >> n;
47     for (int i = 0; i < n; i++)
48     {
49         cin >> num[i][i];
50         if (i > 0) num[i-1][i] = dp[i-1][i] = num[i-1][i-1] + num[i][i];
51     }
52     cout << DPmin(n) << endl << DPmax(n) << endl;
53     return 0;
54 }
55
```

输入

3-2 > input.txt

```
1  4
2  4 4 5 9
```

输出

```
3-2 > output.txt
1 43
2 54
3
```

3-4

代码

```
3-4 > C++ 3-4.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  const int N = 110;
6  // dp[i][j] represents max path from top to i th floor and j th number.
7  // dp[i][j] = max{dp[i-1][j], dp[i-1][j-1]} + num[i, j];
8  int dp[N][N];
9
10 int main()
11 {
12     freopen("input.txt", "r", stdin);
13     freopen("output.txt", "w", stdout);
14     int n, m; cin >> n;
15     cin >> dp[0][0];
16     for(int i = 1; i < n; i++){
17         cin >> m; dp[i][0] = dp[i-1][0] + m;
18         for(int j = 1; j <= i; j++){
19             cin >> m;
20             dp[i][j] = max(dp[i-1][j], dp[i-1][j-1]) + m;
21         }
22     }
23     int res = 0;
24     for(int i = 0; i < n; i++) res = max(res, dp[n-1][i]);
25     cout << res;
26     return 0;
27 }
```

输入

```
3-4 > input.txt
1 5
2 7
3 3 8
4 8 1 0
5 2 7 4 4
6 4 5 2 6 5
```

输出

```
3-4 > output.txt
1 30
```

3-6

代码

```
C++ 3-6.cpp X input.txt 3-6 output.txt 3-6 C++ 3-4.cpp input.txt 3-4 output.txt 3-4 input.txt 3-2 output.txt
3-6 > C++ 3-6.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4  const int N = 110;
5  int price[N][N];
6
7  int main()
8  {
9      freopen("input.txt", "r", stdin);
10     freopen("output.txt", "w", stdout);
11     int n; cin >> n;
12     for(int i = 0; i < n; i++) for(int j = i+1; j < n; j++) cin >> price[i][j];
13
14     for (int j = 2; j < n; j++){
15         for (int i = 0; i + j < n; i++){
16             for(int k = i + 1; k < j; k++){
17                 price[i][j] = min(price[i][k] + price[k][j], price[i][j]);
18             }
19         }
20     }
21     cout << price[0][n-1];
22     return 0;
23 }
```

输入

```
3-6 > input.txt  
1 3 5 15 7
```

输出

```
3-6 > output.txt  
1 12
```

4-1

代码

```
4-1 > C++ 4-1.cpp > main()
1  #include <iostream>
2  #include <algorithm>
3  #include <vector>
4
5  using namespace std;
6  typedef pair<int,int> pii;
7  vector<pii> ori;
8  vector<int> end_time;
9
10 bool my_comp(pii a, pii b){
11     return a.first < b.first;
12 }
13
14 int main()
15 {
16     freopen("input.txt", "r", stdin);
17     freopen("output.txt", "w", stdout);
18     int n; cin >> n; pii tmp;
19     for(int i = 0; i < n; i++){
20         cin >> tmp.first >> tmp.second;
21         ori.push_back(tmp);
22     }
23     sort(ori.begin(),ori.end(), my_comp);
24     for(int i = 0;i<n;i++){
25         bool flag = false;
26         tmp = ori[i];
27         for(int j = 0; j < end_time.size();j++){
28             if(end_time[j] < tmp.first){
29                 end_time[j] = tmp.second;
30                 flag = true;
31                 break;
32             }
33         }
34         if(!flag) end_time.push_back(tmp.second);
35     }
36     cout << end_time.size();
37     return 0;
38 }
39
40
```

输入

```
4-1 > input.txt
1 5
2 1 23
3 12 28
4 25 35
5 27 80
6 36 50
```

输出

```
4-1 > output.txt
1 3
```

4-9

证明

对于选好的分割如果还有更小值，则需要将其中 n 组相邻合为 n_1 个。将每相邻分割的分界线看做一个边。这里看做是对已选定的边进行“平移”，直接去掉某条边看做该边平移至下一条边。显然，每次平移至少需要一条边向后平移。

假设可以平移，则对于向后平移的边，其平移所得到的新分割之和 = 原分割之和 + 原边所在的下一个数且要求新分割之和 $< n$ ，但如果新分割之和 $< n$ ，原边就并不符合我们贪心的算法。

代码

```
4-9 > 4-9.cpp x output.txt 4-9 4-1.cpp input.txt 4-1 output.txt 4-1
4-9 > 4-9.cpp > main()
1  #include <iostream>
2  #include <algorithm>
3  #include <vector>
4
5  using namespace std;
6
7  int main()
8  {
9      freopen("input.txt", "r", stdin);
10     freopen("output.txt", "w", stdout);
11     int n,k; cin >> n >> k;
12     int tmp_n = n, num = 0;
13     for(int i = 0;i < k+1;i++){
14         int tmp; cin >> tmp;
15         if(tmp > n){
16             cout << "No Solution";
17             return 0;
18         }
19         if(tmp_n > tmp) tmp_n -= tmp;
20         else tmp_n = n - tmp, num++;
21     }
22     cout << num;
23     return 0;
24 }
```

输入

```
4-9 > input.txt
1  7 7
2  1 2 3 4 5 1 6 6
```


输出

```
4-9 > output.txt
1 4
```

4-11

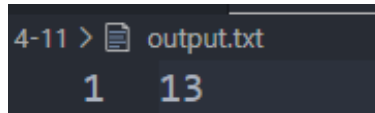
代码

```
C++ 4-11.cpp x
4-11 > C++ 4-11.cpp > main()
1  #include <iostream>
2  #include <algorithm>
3  #include <vector>
4
5  using namespace std;
6
7
8  int main()
9  {
10     freopen("input.txt", "r", stdin);
11     freopen("output.txt", "w", stdout);
12     string ori; int n; cin >> ori >> n;
13     for(int i = 0; i < n; i++)
14         ori.erase(max_element(ori.begin(), ori.end()));
15     cout << ori;
16     return 0;
17 }
```

输入

```
4-11 > input.txt
1 178543
2 4
```

输出

A terminal window with a dark background. The first line shows a prompt '4-11 >' followed by a file icon and the text 'output.txt'. The second line shows the numbers '1' and '13' separated by a space.

Other things

\LaTeX code refer to these things and was compiled on texlive2020.

- [UCB-CS70's given homework template.](#)
- [A free website useful to edit \$\text{\LaTeX}\$ formula code.](#)

Thanks for your correcting and grading :).