

HW3

activity-selection problem

b10532027 四電資二 林科廷

程式摘要：

由 C 語言撰寫, 將輸入的活動, 依照結束的時間, 由小到大作排序, 再依 rod-cutting 的方式建表

程式本體 & 註解：

```
1. #include <iostream>
2. #include <vector>
3. #include <algorithm>
4.
5. using namespace std;
6.
7. class activity
8. {
9. public:
10.     int number;
11.     int start;
12.     int end;
13.     int value;
14.     int pre_act;
15.
16.     activity() //活動的資料
17.     {
18.         number = -1;
19.         start = -1;
20.         end = -1;
21.         value = -1;
22.         pre_act = -1;
23.     }
24.     activity(int a, int b, int c, int d)
```

```
25.     {
26.         number = a;
27.         start = b;
28.         end = c;
29.         value = d;
30.         pre_act = -1;
31.     }
32.     bool compatible(activity a)
33.     {
34.         if (a.start >= end || a.end <= start)
35.         {
36.             return true;
37.         }
38.         else return false;
39.     }
40. };
41.
42. bool comp(activity a, activity b)
43. {
44.     return a.end < b.end;
45. } //依照結束時間做排序
46. bool comp_ans(int a, int b)
47. {
48.     return a < b;
49. } //依照結束時間做排序
50.
51.
52.
53. int main()
54. {
55.     int n;
56.     cin >> n;
57.
58.
59.     for (int i = 0; i < n; i++)
60.     {
61.         activity dp[999];
62.         vector<activity>activities;
```

```

63.
64.     int m;
65.     cin >> m;
66.
67.     for (int j = 0; j < m; j++)//讀測資
68.     {
69.         int temp1, temp2, temp3, temp4;
70.         cin >> temp1 >> temp2 >> temp3 >> temp4;
71.         if(temp2<=temp3)
72.             activities.push_back(activity(temp1, te
mp2, temp3, temp4));
73.     }
74.
75.     sort(activities.begin(), activities.end(), comp
); //依照結束時間做排序
76.
77.     for (int j = 0; j < m; j++)//初始化
78.     {
79.         dp[j] = activities[j];
80.     }
81.
82.     for (int k = 1; k < m; k++)
83.     {
84.
85.         for (int j = 0; j < k; j++)
86.         {
87.
88.             if (activities[k].compatible(activities
[j]) && dp[j].value + activities[k].value > dp[k].value)/
/建表
89.             {
90.                 dp[k].value = dp[j].value + activit
ies[k].value;
91.                 dp[k].pre_act = dp[j].number;
92.             }
93.         }
94.     }
95.

```

```

96.
97.
98.     int max_ans = -1, max_value = -1;
99.     for (int k = 0; k < m; k++)
100.    {
101.        if (dp[k].value > max_value)
102.        {
103.            max_value = dp[k].value;
104.            max_ans = k;
105.        }
106.    }
107.
108.
109.    vector<int>ans;
110.    cout << dp[max_ans].value << endl;
111.    ans.push_back(dp[max_ans].number);
112.    activity temp = dp[max_ans];
113.
114.
115.
116.    for (int k = 0; k < m ; k++)
117.    {
118.        //cout <<k<<" "<< dp[k].number <<" "<< temp
        .pre_act << endl;
119.        if (dp[k].number == temp.pre_act)
120.        {
121.            ans.push_back(temp.pre_act);
122.            temp = dp[k];
123.            k = -1;
124.        }
125.    }
126.
127.    sort(ans.begin(),ans.end(),comp_ans);
128.    for (int k = 0; k < ans.size(); k++)
129.    {
130.        cout << ans[k];
131.        if (k != ans.size() - 1)
132.            cout << " ";

```

```

133.     }
134.     cout << endl;
135.
136. }
137.
138.
139.     //system("pause");
140.     return 0;
141. }

```

Pseudo code:

Sort(activity, finishtime)

Initializing table T

For I from 1 to n

 For j from 0 to i-1

 If(compatible)

$T[i] = \max(T[i], T[j] + \text{value}[i])$ 圖解 :

圖解 :

