

# 計算機韌體實驗 (P14) 木塊問題/The Blocks Problem

饒建奇 (Jiann-Chyi Rau)  
淡江大學電機工程學系，E627  
E-mail: [jcrau@ee.tku.edu.tw](mailto:jcrau@ee.tku.edu.tw)

## 解題要訣 (1/2)

- 每個木塊堆的高度不確定
  - 用 vector 來儲存
- 木塊堆的數目依題意不超過25

```
20 vector<int> pipe[25]; //二維陣列(第一維大小根據題意, 第二維大小不固定)
```

## 解題要訣 (2/2)

- move a = 將a上方的木塊歸位 + pipe a
- onto b = 將b上方的木塊歸位 + over b

```
93 //把第p堆高度為h的木塊"上方的"所有木塊歸位
94 void clear_above(int p, int h)
95 {
96     int i;
97     int b;
98
99     for(i=pipe[p].size()-1; i>h; i--){
100         b = pipe[p][i];
101         pipe[b].push_back(b);
102     }
103     pipe[p].resize(h+1);
104 }
105
```

## 發現木塊的位置與高度

```
81 //發現木塊a的位置p和高度h(皆從0算起)
82 void find_block(int a, int &p, int &h)
83 {
84     for(p=0; p<n; p++){
85         for(h=0; h<pipe[p].size(); h++){
86             if(pipe[p][h] == a){
87                 return;
88             }
89         }
90     }
91 }
```

# pipe a over b

```
106 //把第p堆高度h"及其上方"的木塊整體移動到第p2堆的頂部
107 void pipe_over(int p, int h, int p2)
108 {
109     int i;
110
111     for(i=h; i<pipe[p].size(); i++){
112         pipe[p2].push_back(pipe[p][i]);
113     }
114     pipe[p].resize(h);
115 }
```

# 模擬木塊操作指令

```
43         cin >> a >> s2 >> b;
44
45         find_block(a, pa, ha);
46         find_block(b, pb, hb);
47
48         if(pa == pb){
49             continue; //非法指令
50         }
51         if(s1 == "move"){
52             clear_above(pa, ha);
53         }
54
55         if(s2 == "onto"){
56             clear_above(pb, hb);
57         }
58
59         pipe_over(pa, ha, pb);
```

## 輸出所有的木塊堆

```
128 //輸出結果
129 void show(void)
130 {
131     int i, j;
132
133     for(i=0; i<n; i++){
134         printf("%d: ", i);
135         for(j=0; j<pipe[i].size(); j++){
136             printf("%d ", pipe[i][j]);
137         }
138         printf("\n");
139     }
140 }
```