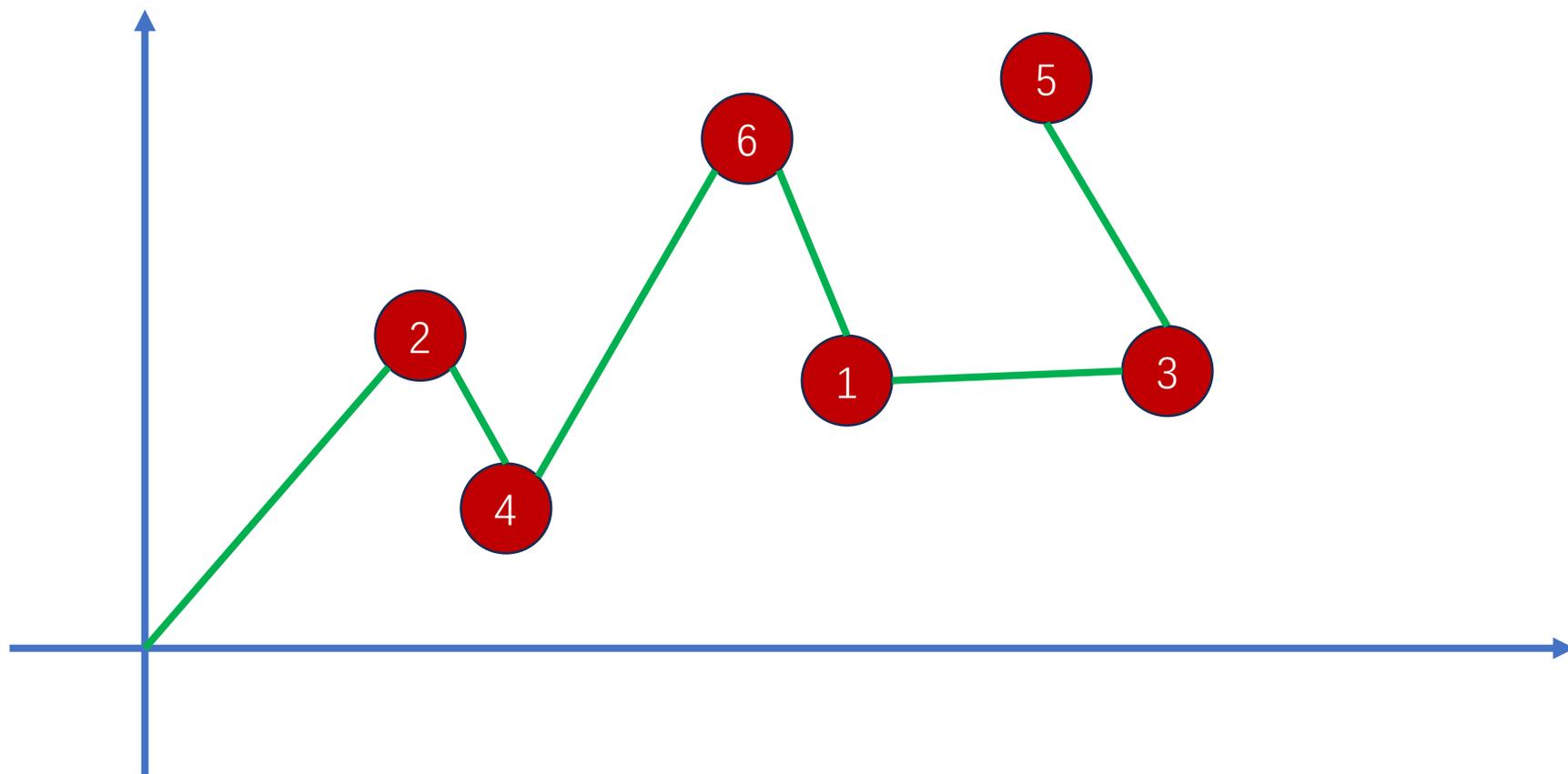


第六讲：吃奶酪-P1433

胡船长

初航我带你，远航靠自己

一、问题搜索树



一、问题搜索树

2 (1, 2)

4 (2, 4)

6 (3, 6)

1 (4, 1)

3 (5, 3)

5 (6, 5)

排列型枚举问题

二、优化1：状态表示

● (1, 2)

● (2, 4)

●

●

●

●

t =

6	5	4	3	2	1	0
1	1	0	1	0	1	0

三、优化2：搜索过程剪枝

(1 , 2) (1 , 4)

(2 , 4) (2 , 2)

(3 , 6) (3 , 6)

1、 3、 5

三、优化2：搜索过程剪枝

100

(1 , 2)

(2 , 4)

(3 , 6)

10

(1 , 4)

(2 , 2)

(3 , 6)

1、 3、 5

三、优化2：搜索过程剪枝

100	10
(1 , 2)	(1 , 4)
(2 , 4)	(2 , 2)
(3 , 6)	(3 , 6)

6	5	4	3	2	1	0
0	1	0	1	0	1	0

1、 3、 5

三、优化2：搜索过程剪枝

100	10
(1 , 2)	(1 , 4)
(2 , 4)	(2 , 2)
(3 , 6)	(3 , 6)

1、 3、 5

1. $dp[t][6] = 100$
2. $dp[t][6] = 10$ 可以继续搜索

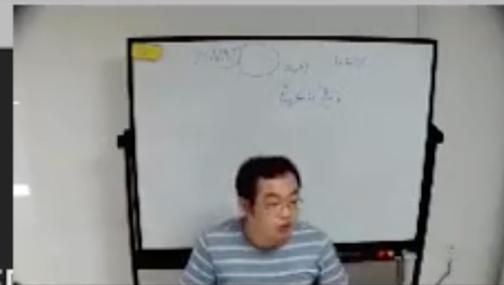
三、优化2：搜索过程剪枝

100	10
(1 , 2)	(1 , 4)
(2 , 4)	(2 , 2)
(3 , 6)	(3 , 6)

1、 3、 5

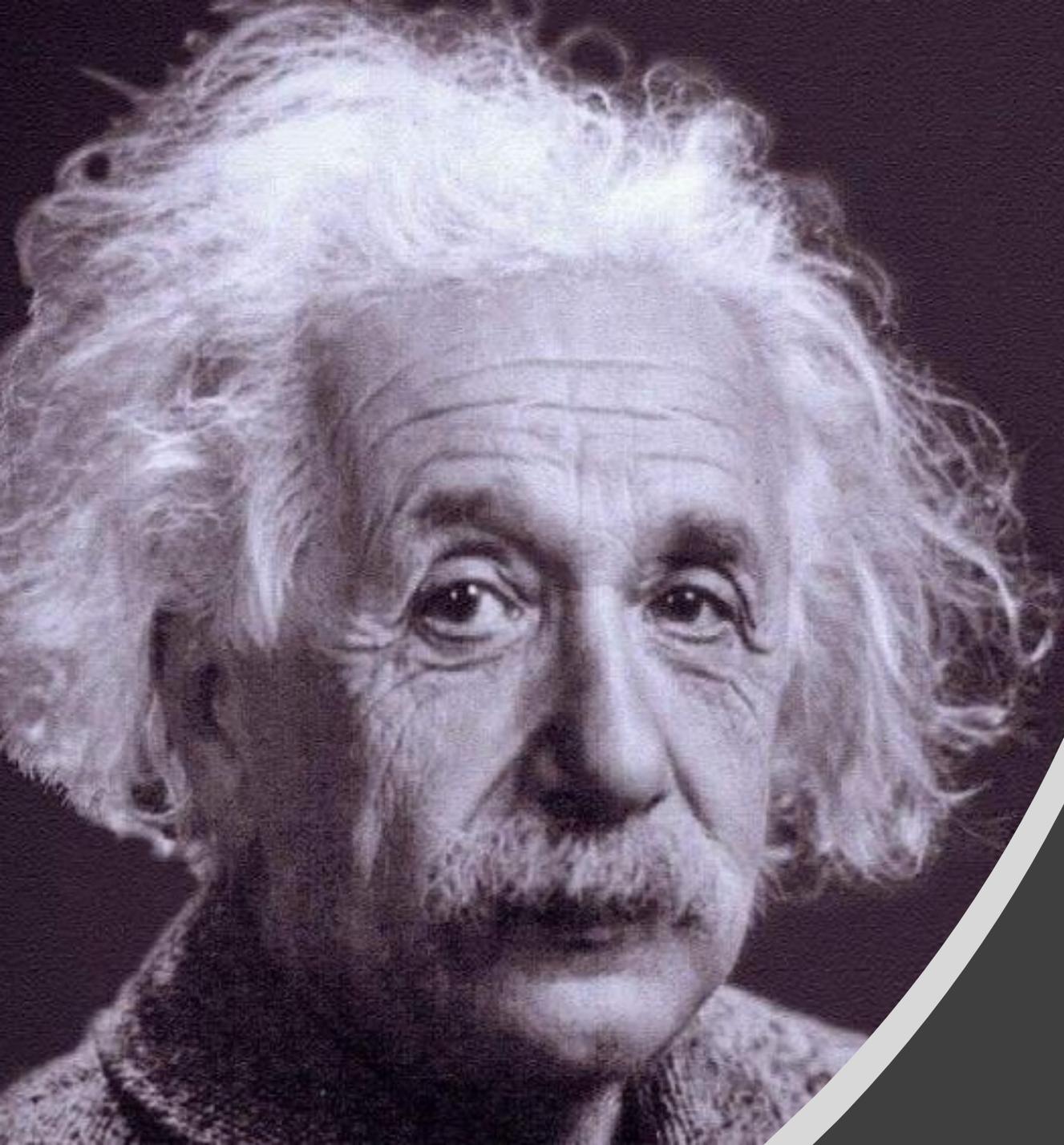
1. $dp[t][6] = 10$
2. $dp[t][6] = 100$ 可以不用搜了

```
39 }
40
41 Node *insert_maintain(Node *root) {
42     if (!hasRedChild(root)) return root;
43     if (root->lchild->color == RED && root->rchild->color == RED, {
44         if (!hasRedChild(root->lchild) && !hasRedChild(root->rchild)) return root;
45         root->color = RED;
46         root->lchild->color = root->rchild->color = BLACK;
47         return root;
48     }
49     if (root->lchild->color == RED) {
50         if (!hasRedChild(root->lchild)) return root;
51
52     } else {
53         if (!hasRedChild(root->rchild)) return root;
54
55     }
56 }
57
```



吃奶酪-P1433：代码演示

```
61 Node *__insert(Node *root, int key) {
62     if (root == NIL) return getNewNode(key);
```



为什么
会出一样的题目？