



更多排列问题

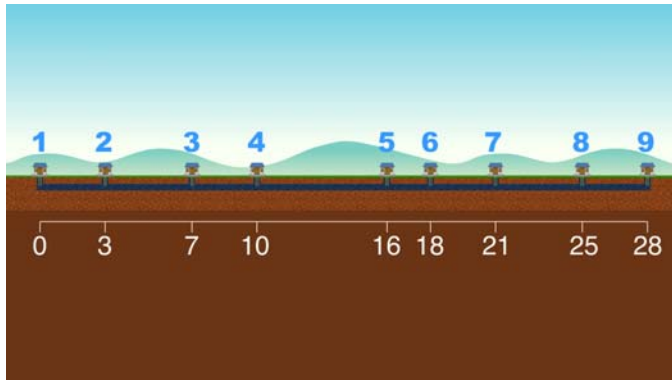
李浩文、彼得·斯塔基



秘道传信

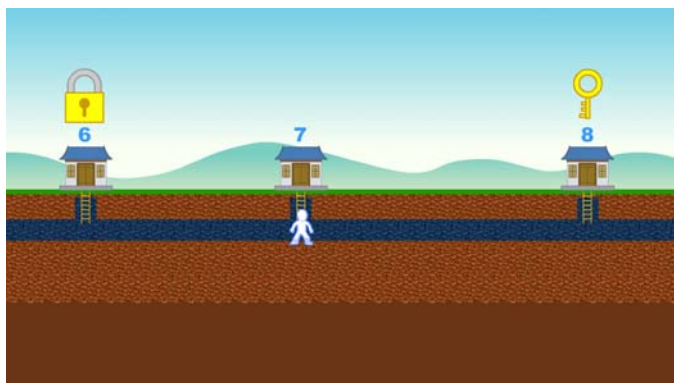


秘道问题



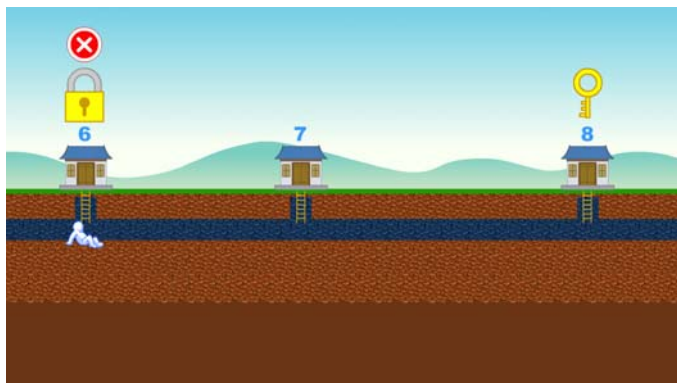
3

地点优先次序



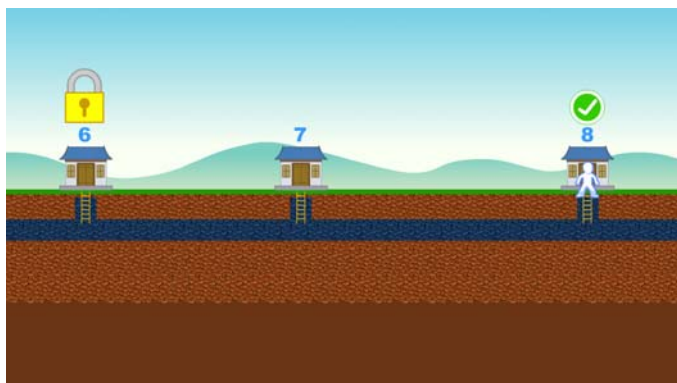
4

地点优先次序



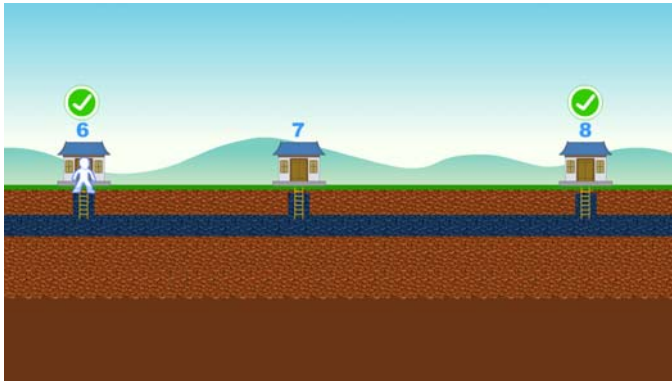
5

地点优先次序



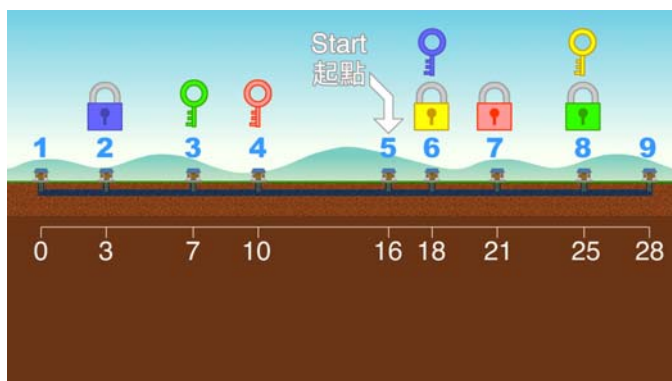
6

地点优先次序



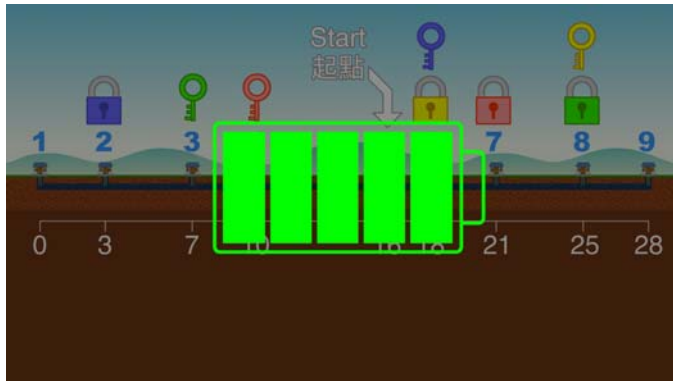
7

地点优先次序



8

目标



9

直线上的旅行商问题

- 给定一个位于一条直线上的军事中枢点集合以及这些中枢点的优先次序集合，从**第五个**中枢点**开始**依次访问各个中枢点从而
 - 满足中枢点优先次序，并且
 - 最小化移动的总距离

10



排列问题

- 排列问题是一类重要的匹配问题
 - 按某一顺序放置一组 OBJ 对象
- 这是 OBJ 与 $1..n$ 的一个匹配
 - n 是 OBJ 的势
- 至少有两个视角
 - DOM = OBJ 和 COD = $1..n$
 - DOM = $1..n$ 和 COD = OBJ
- 衣带密诏问题和这个秘道问题都是排列问题

11

秘道问题模型 (tunnel.mzn)

数据

```
enum PIVOT;  
PIVOT: first;  
  
set of int: POS = 1..card(PIVOT);  
array[PIVOT] of int: coord; % coord of pivot  
  
int: m; % number of precedences  
set of int: PREC = 1..m;  
array[PREC] of PIVOT: left;  
array[PREC] of PIVOT: right;
```

12

秘道问题模型 (tunnel.mzn)

两个视角的决策变量

- **order**: 每个中枢点的位置 (顺序)

```
array[PIVOT] of var POS: order;
```

- **route**: 路线中每步所到的中枢点

```
array[POS] of var PIVOT: route;
```

约束

```
route[1] = first;  
inverse(order,route);  
forall(i in PREC)  
    (order[left[i]] < order[right[i]]);
```

目标

```
solve minimize sum(i in 1..card(PIVOT)-1)  
    (abs(coord[route[i]] - coord[route[i+1]]));
```

13

对模型求解

```
Route: [P5, P4, P3, P9, P8, P7, P6, P2, P1]
```

```
Total Distance: 58
```

14



小结

- ❏ **秘道**问题是计算机科学中典型的旅行商问题的一个简化，但我们的例子中还有额外约束（优先次序）
- ❏ 旅行商问题是一个重要的图论问题，并通常在路线规划和优化问题具有重要应用
- ❏ 在我们的例子中，一些需求**无法**在某个特定视角下表示；这时就只能利用结合模型来阐述整个问题

15

小结

- ❏ 排列问题
 - 总是有至少两个视角
- ❏ 选用视角使之：
 - **可以／容易地** 表示约束和目标
- ❏ 否则，选用两个视角并添加
 - **inverse** 约束

16



图像引用

所有图像由Marti Wong设计提供, © 香港中文大学与墨尔本大学 2016

17