

A low-angle, upward-looking photograph of several modern skyscrapers. The buildings feature glass facades and white structural elements, creating a sense of height and architectural complexity. The sky is a clear, vibrant blue. A dark green horizontal band is superimposed across the middle of the image, containing the chapter title.

第三章

问题规约

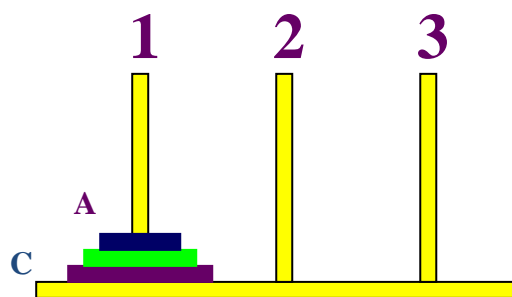


问题规约表示

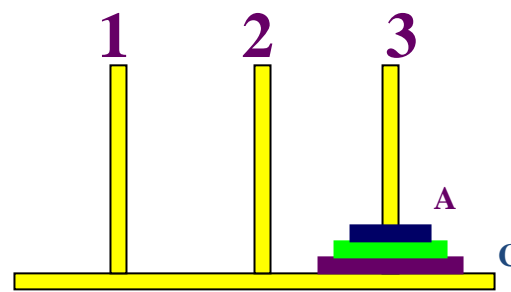


湖北大学
HUBEI UNIVERSITY

- 梵塔问题 (tower of Hanoi puzzle)
 - ✓ 有3个柱子，以及 $n=3$ (问题的规模) 个尺寸各不相同的圆盘
 - ✓ 最初所有圆盘都在柱子1上，目标是将圆盘全部移动到柱子3上
 - ✓ **要求：**每次只能移动一个圆盘；只允许移动柱子顶端的圆盘；不允许将较大的圆盘叠放在较小的圆盘上



(a) 初始状态



(b) 目标状态



问题规约表示



湖北大学
HUBEI UNIVERSITY

- 问题规约 (problem reduction)

- ✓ 需要解决：一个规模为 n 的问题
- ✓ 能否找到一种解法，且这种解法形如：

解决若干个规模更小的（如规模为 $n-1$ ）同类问题

+

解决若干个平凡问题 (trivial problem)

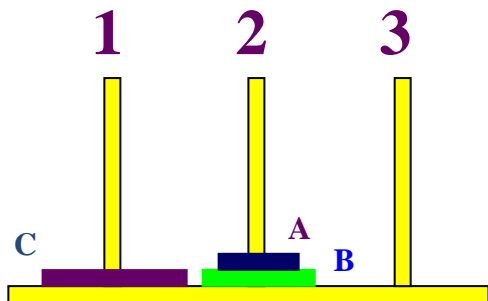


问题规约表示

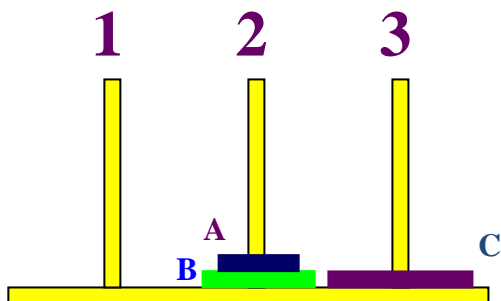


湖北大学
HUBEI UNIVERSITY

- 梵塔问题 (tower of Hanoi puzzle): 移动 n 个盘子



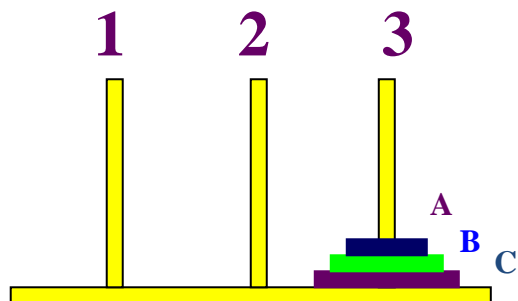
- ✓ 步骤一: 将上层的 $(n-1)$ 个盘子从柱子1移动到柱子2



- ✓ 步骤二: 将最大的盘子移动到柱子3

平凡问题

规模更小的同类问题



- ✓ 步骤三: 将上层的 $(n-1)$ 个盘子从柱子2移动到柱子3

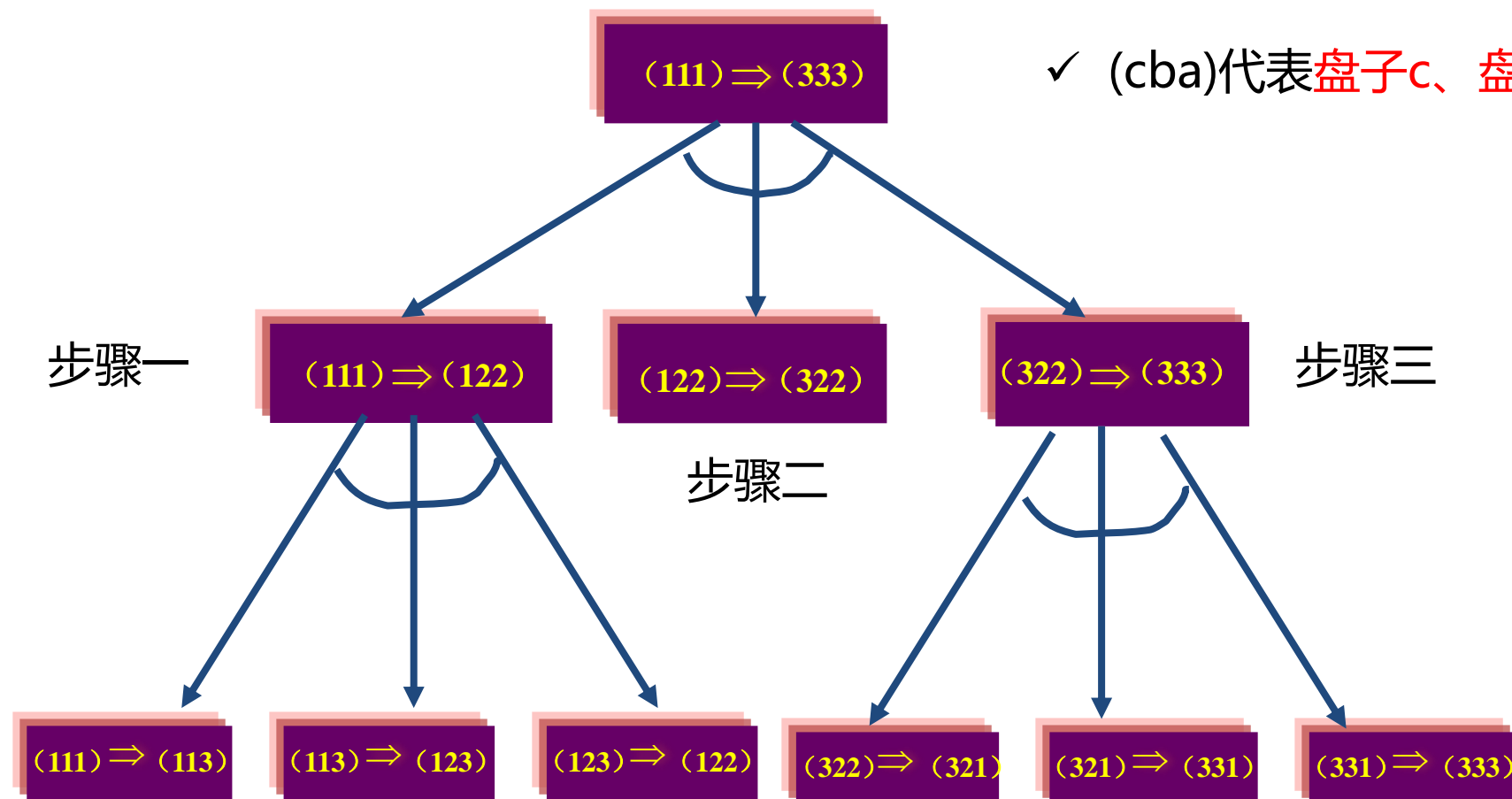


与或图



湖北大学
HUBEI UNIVERSITY

• 梵塔问题的与或图





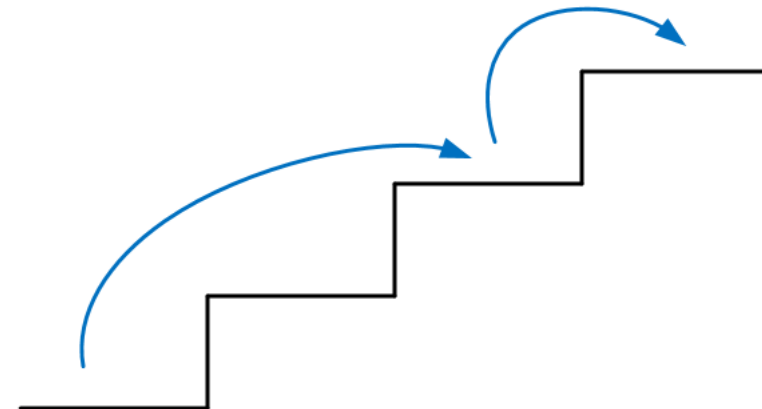
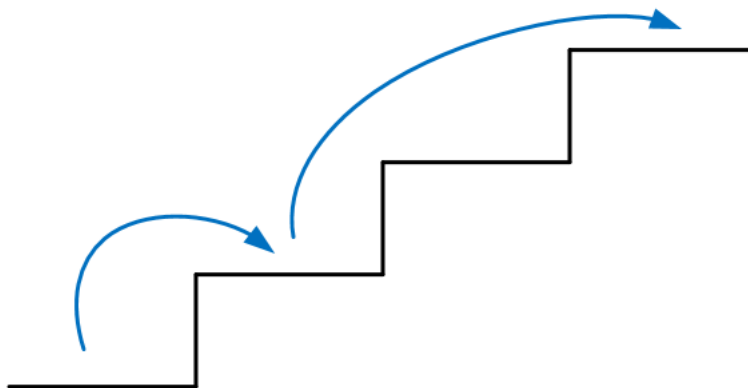
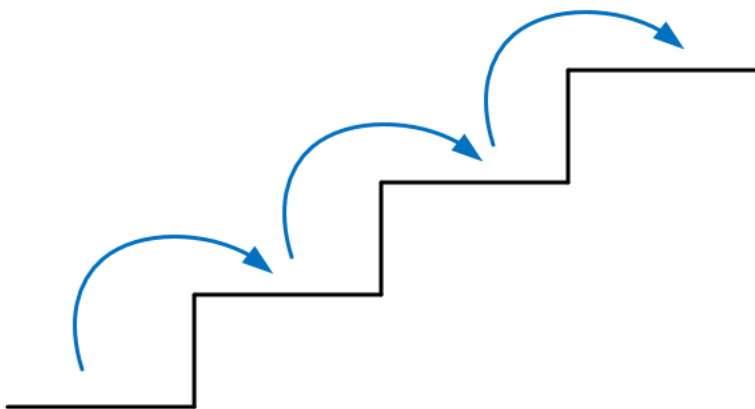
问题规约表示



湖北大学
HUBEI UNIVERSITY

- 爬楼梯问题

- ✓ **问题：**需要爬上 n 级台阶
- ✓ **限制：**每次可以爬1级台阶或2级台阶
- ✓ 问总共有多少种爬完 n 级台阶的方法？



例：想爬上3级台阶，共有3种方法



问题规约表示

规模更小的同类问题

平凡问题

原问题

如何爬上
第n级台阶

先爬完前(n-1)级台阶

最后一次爬1级台阶

先爬完前(n-2)级台阶

最后一次爬2级台阶

- 假设爬完n级台阶总共有 $S(n)$ 种方法

✓ $S(n) = S(n-1) + S(n-2)$

✓ $S(1) = 1$

✓ $S(2) = 2$



湖北大学
HUBEI UNIVERSITY



问题规约表示

- 爬楼梯问题的python实现（使用递归）：输入台阶数，返回解法数量

```
def climb_stairs(n: int) -> int:
    if n <= 0:
        return -1
    if n == 1:
        return 1
    if n == 2:
        return 2
    return climb_stairs(n-1) + climb_stairs(n-2)

print("Number of solutions", climb_stairs(5))
```



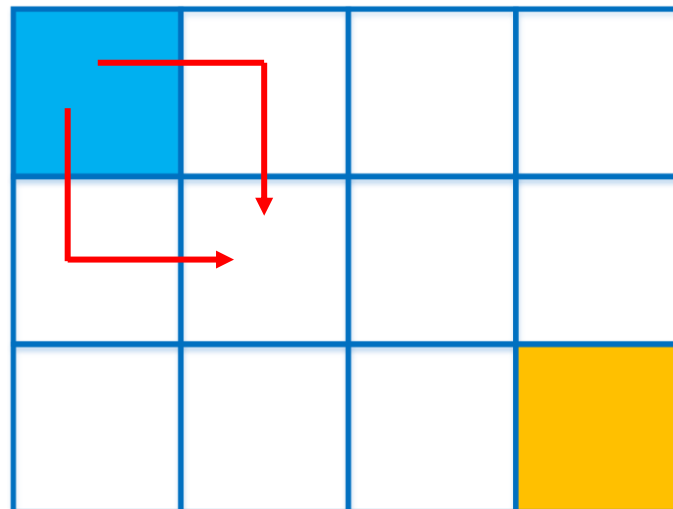

问题规约表示



湖北大学
HUBEI UNIVERSITY

• 路径问题

- ✓ **问题：**从大小 $m \times n$ 为二维数组的左上角出发，移动到右下角
- ✓ **限制：**每次移动1格，只能向右或者向下移动
- ✓ 问总共有多少条不同的路径？



例：想移动到第2排第2列的格子，共有2条路径



问题规约表示

规模更小的同类问题

平凡问题



湖北大学
HUBEI UNIVERSITY

原问题

如何移动到
二维数组 M 的右下角

先移动到 $M[m-2][n-1]$, 最后向下移动1格

先移动到 $M[m-1][n-2]$, 最后向右移动1格

- 假设大小 $m \times n$ 为二维数组, 从左上角移动到右下角
总共有 $S(m, n)$ 方法
 - ✓ $S(m, n) = S(m-1, n) + S(m, n-1)$
 - ✓ $S(1, 1) = 1$



问题规约表示

- **python实现（使用递归）**：输入二维数据的大小，返回路径数

```
def matrix_path(m: int, n: int) -> int:
    if m <= 0 or n <= 0:
        return -1
    if m == 1 and n == 1:
        return 1
    if m == 1 and n > 1:
        return matrix_path(m, n - 1)
    if m > 1 and n == 1:
        return matrix_path(m - 1, n)

    return matrix_path(m - 1, n) + matrix_path(m, n - 1)
```



```
print('Number of path: ', matrix_path(4, 3))
```

结束语



谢谢!