1. 顺序查找的时间复杂O(n)，二分查找的时间复杂度为O(logn)，将n = 1000000代入比值为1000000 / log2(1000000) = 50171.665944倍

考虑平均比较次数

顺序查找：

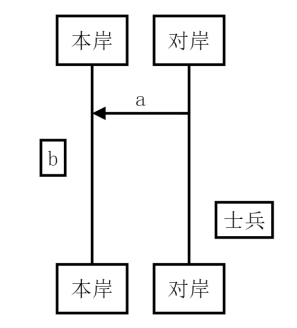
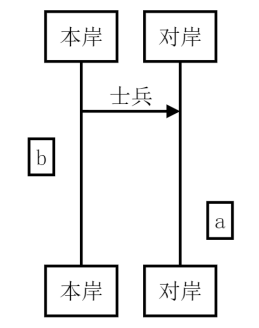
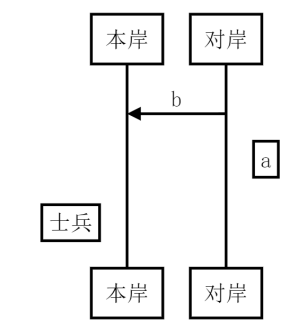
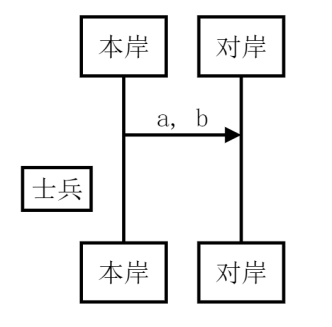
折半查找：

带入n = 1000000，结果为：26410.91倍

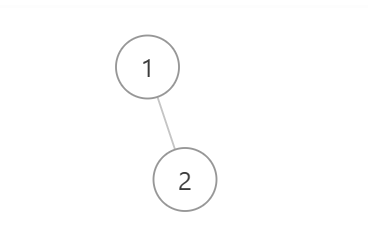
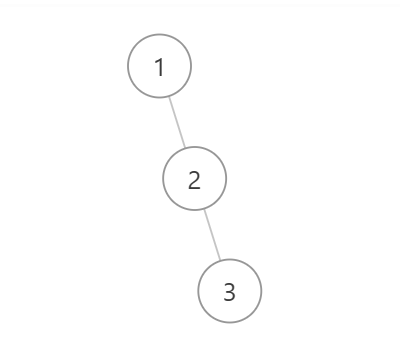
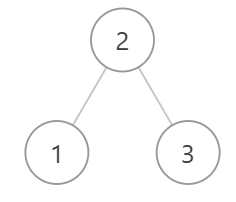
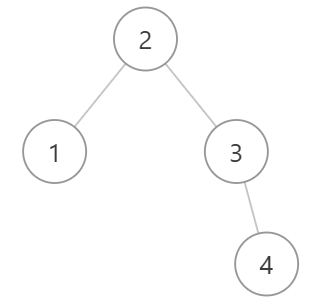
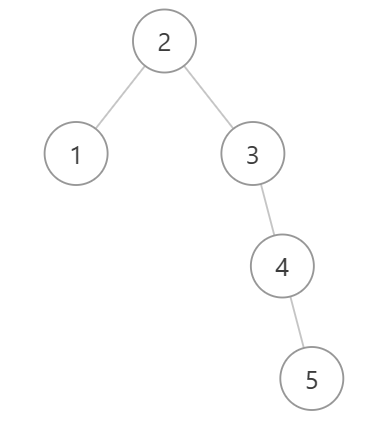
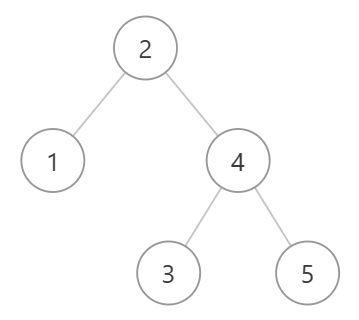
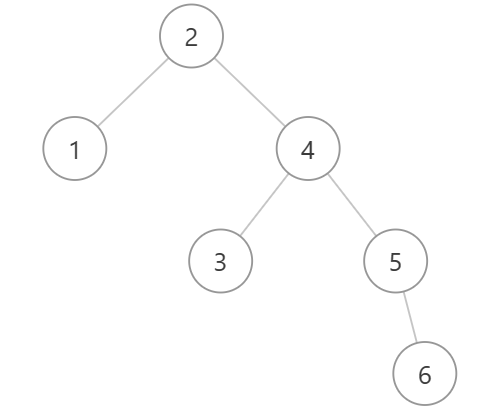
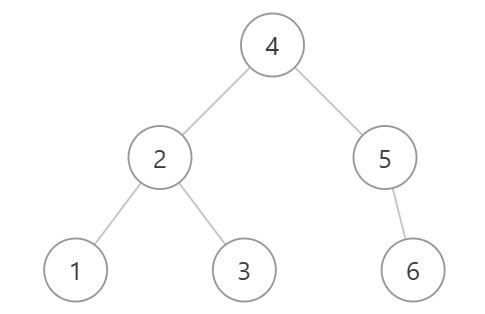
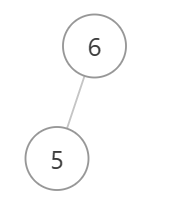
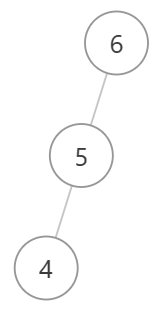
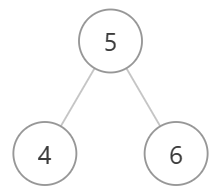
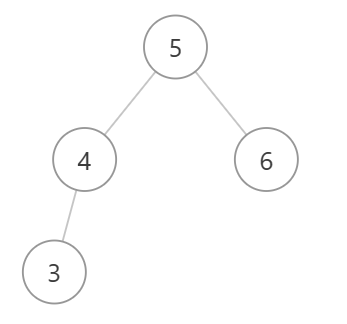
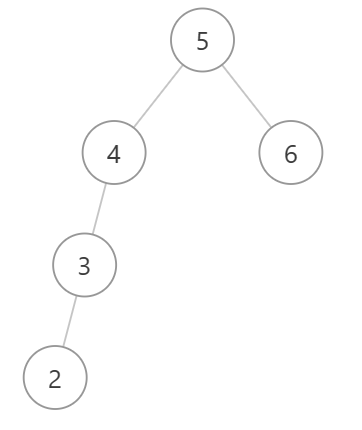
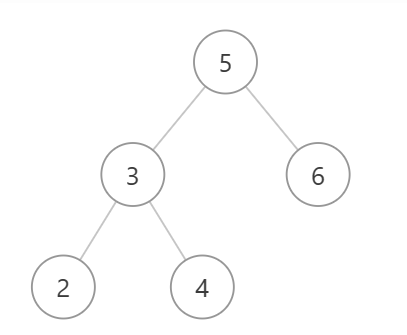
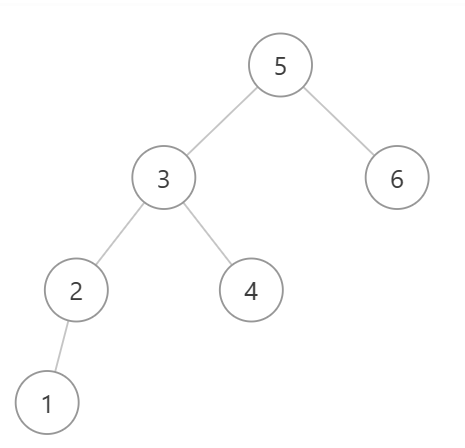
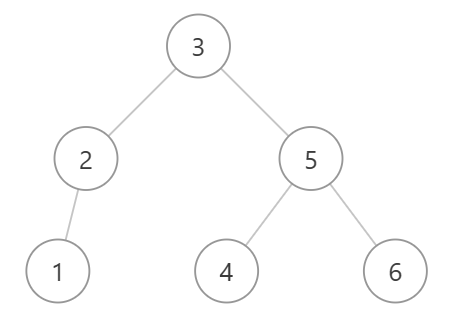
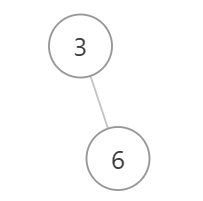
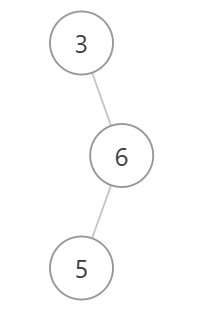
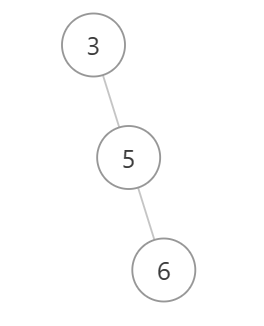
差异分析：在时间复杂度表示中忽略了系数，因此相差几乎两倍

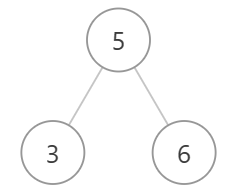
1. 运送士兵过河
   1. 方法，令两个小男孩为a、b，过河士兵为s1, s2, ..., sn
      1. 首先让小男孩a、b过河，选择任意一个小男孩留在对岸，这里不妨选择a
      2. 然后b回到本岸下船
      3. 让士兵si(1 <= i <= n)过河到对岸，士兵si下船
      4. 然后小男孩a坐船回到本岸

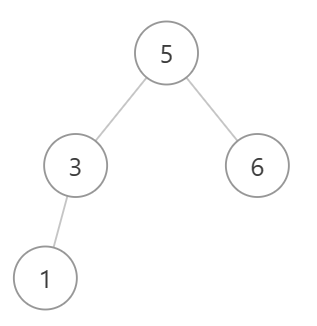
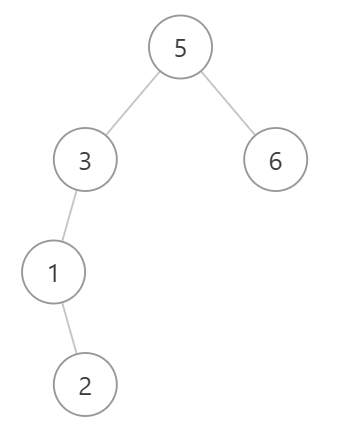
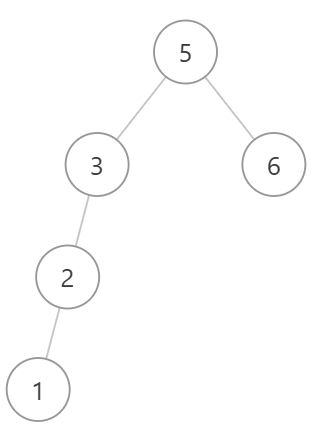
示意图如下：

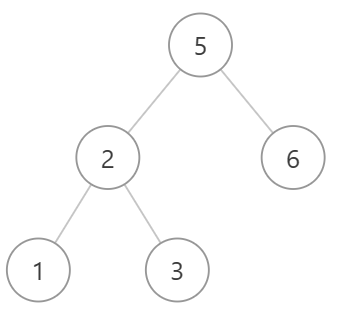


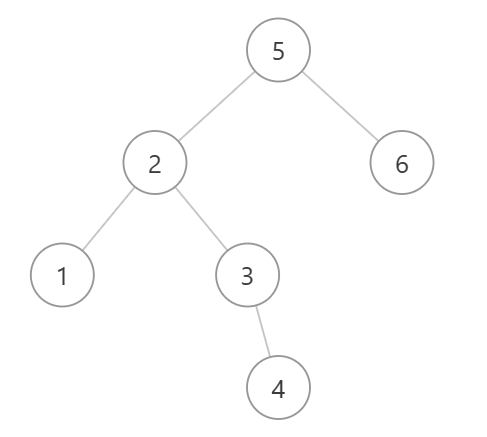
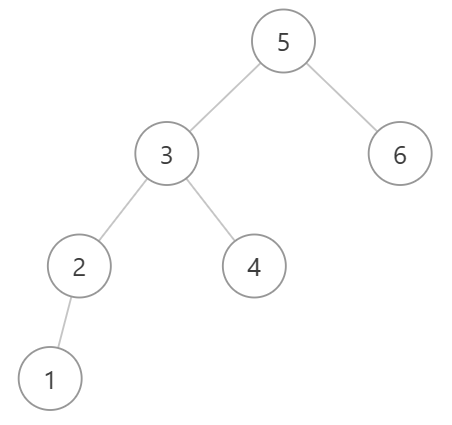
* 1. 综上，运送一个士兵需要四次岸到岸的横渡，于是对于N个士兵，需要4N次横渡

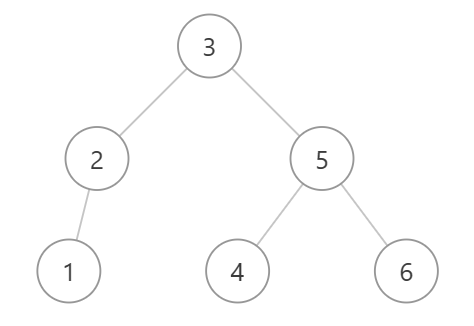
1. 遍历顺序
   1. 前序遍历abdecf
   2. 中序遍历dbeacf
   3. 后序遍历debfca
2. 构造AVL树
   1. 123456
      1. 插入1
      2. 插入2
      3. 插入3需要调整为
      4. 插入4
      5. 插入5需要调整为
      6. 插入6需要调整为
   2. 654321
      1. 插入6
      2. 插入5
      3. 插入4需要调整为
      4. 插入3
      5. 插入2需要调整为
      6. 插入1需要调整为
   3. 365124
      1. 插入3
      2. 插入6
      3. 插入5调整为

再调整为

* + 1. 插入1
    2. 插入2调整为

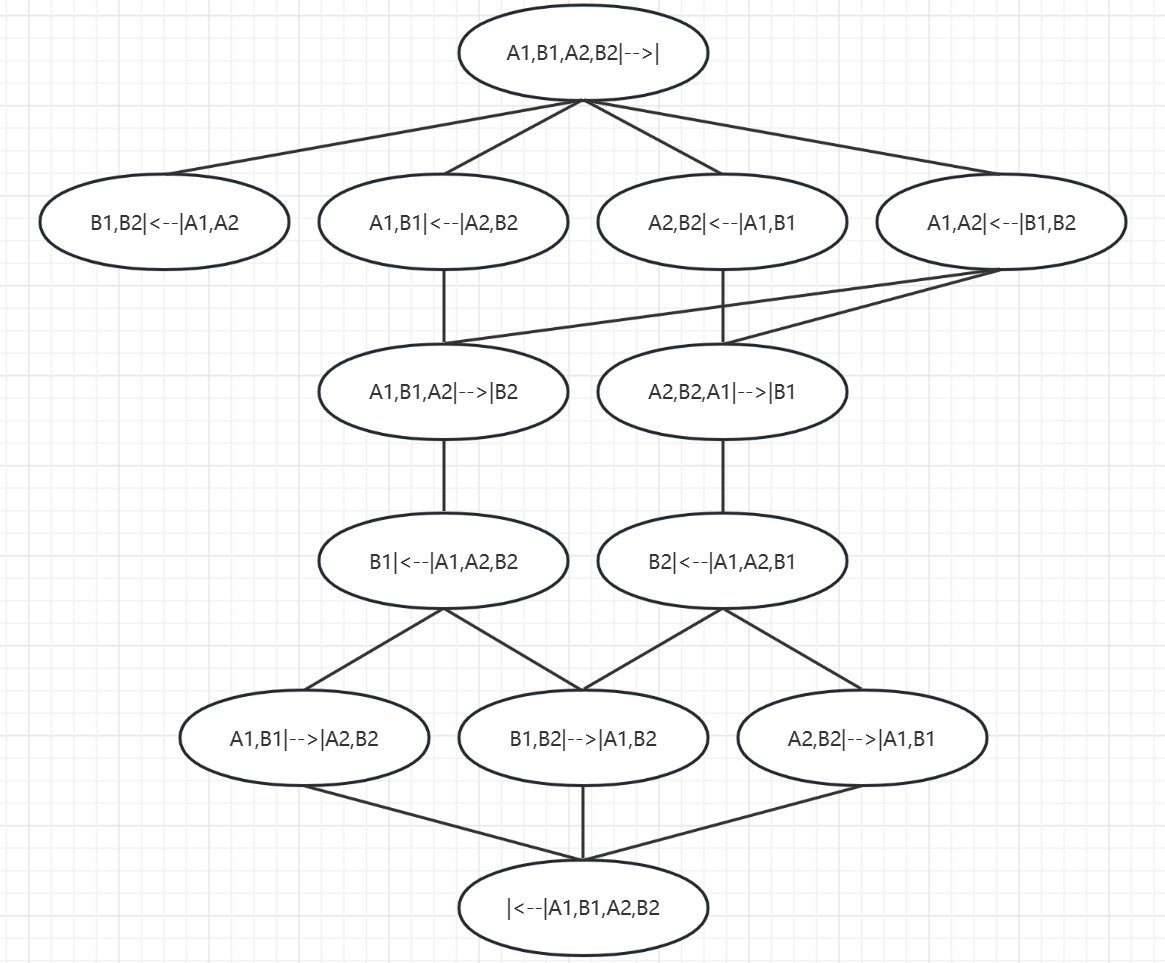
再调整为

* + 1. 插入4调整为

再调整为

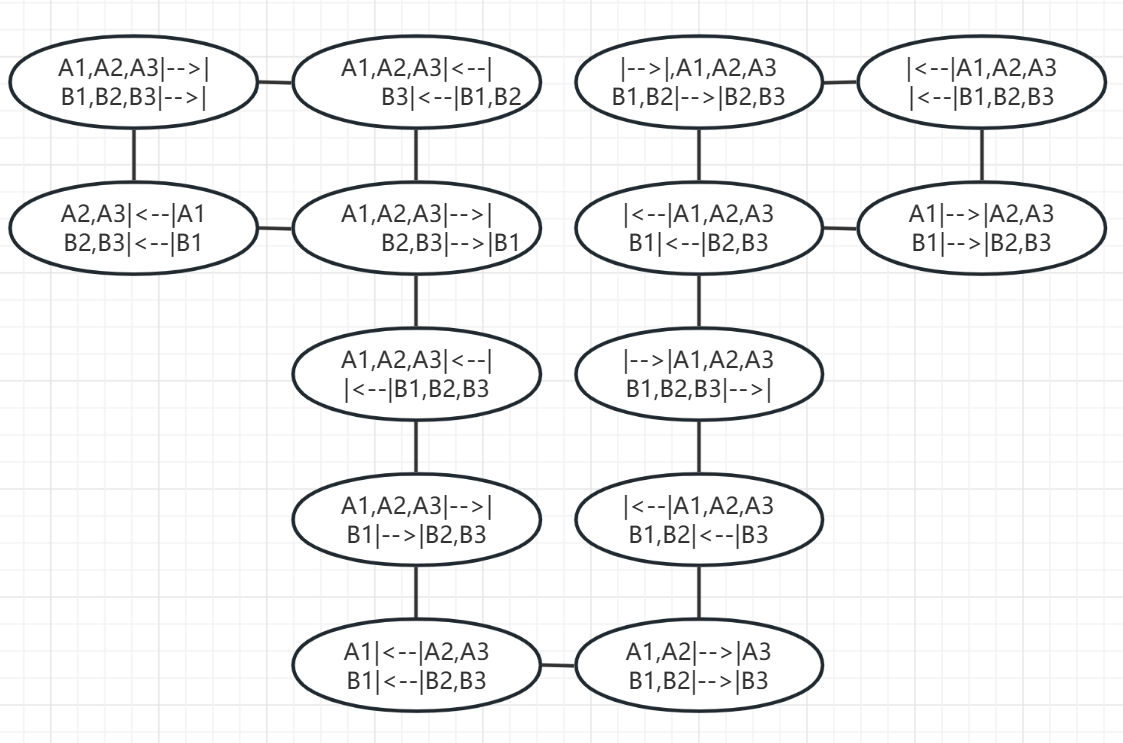
1. 爱吃醋的丈夫
   1. n = 2，不妨令丈夫为Ai，妻子为Bi，共需要5次渡河

状态图如下：



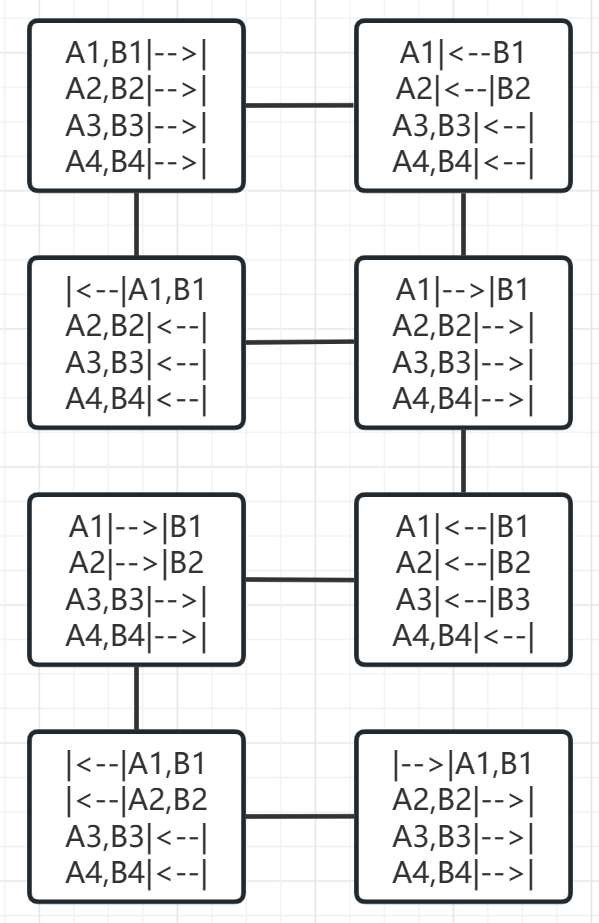
* 1. n = 3，不妨令丈夫为Ai，妻子为Bi，共需要11次渡河

状态图如下：



* 1. n = 4,该问题n≥4时无解。在上一小问3对夫妻的基础上，从1个或多个额外的夫妇开始，将不会产生新的状态，并且在六次过河之后，将到达n-1对夫妇和船在初始河岸，有一对夫妇在另一岸的状态，从该状态唯一允许的变化是将该夫妇运送到另一岸而回到之前的状态。

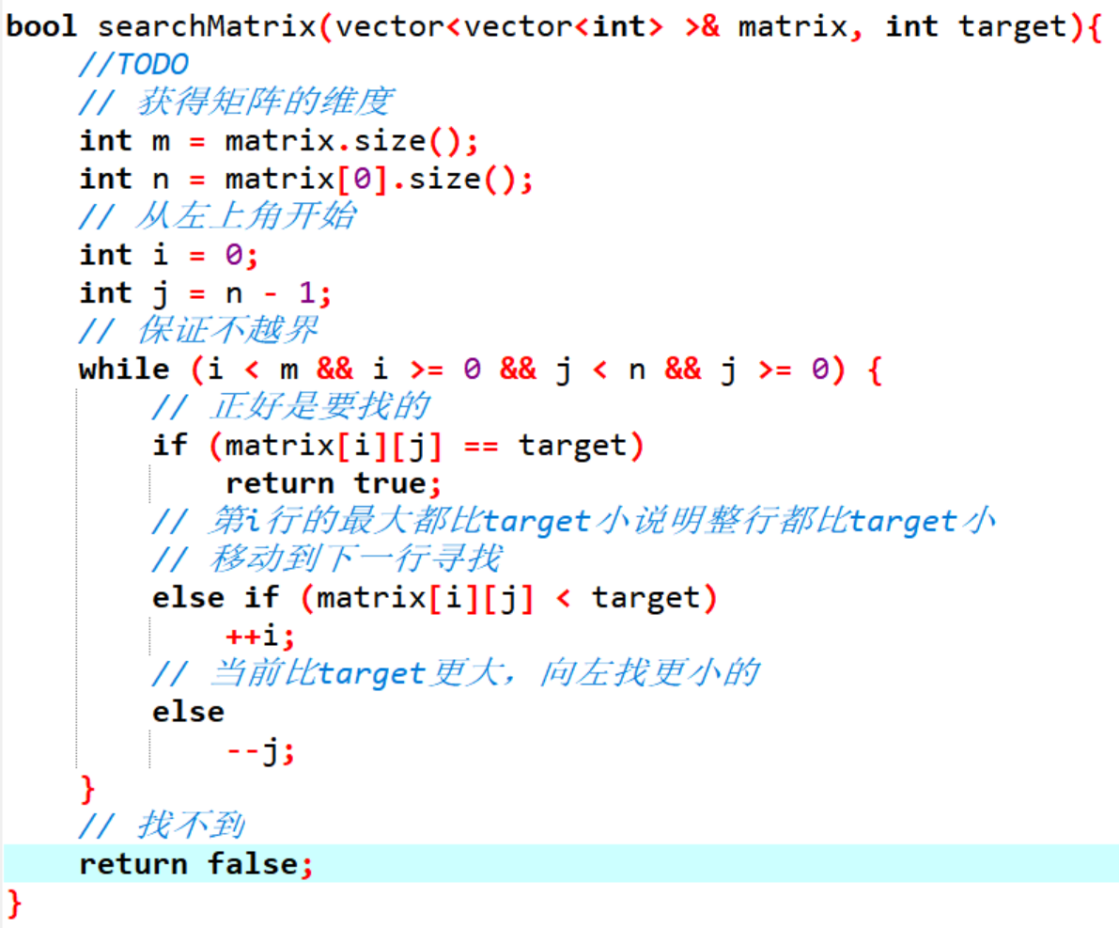
状态图如下：



* 1. 算法思路

1. 从矩阵的右上角即matrix[0][n-1]开始搜索，每一轮搜索中当前位置为matrix[i][j]
2. 若matrix[i][j] == target，说明搜索成功
3. 若matrix[i][j] < target，说明该target大于该行左侧的所有数字，忽略，则执行i <- i + 1，到下一行
4. 若matrix[i][j] > target，说明target小于该列上侧的所有数字，忽略，则执行j <- j - 1，到左边一列
5. 重复上述步骤直到到达边界为止
   1. 时间复杂度分析
      1. 输入规模：矩阵行数m、矩阵列数n
      2. 基本操作：比较
      3. 检查
         1. 最好情况：target == matrix[0][n - 1]，时间复杂度为O(1)
         2. 最坏情况：i增加m次，j减少n次，时间复杂度为O(m + n)

④ 时间复杂度为O(m + n)

* 1. 空间复杂度分析
     1. 使用常数个数的变量，空间复杂度为O(1)
  2. 补充的函数
  3. 运行截图

