

1.请估计一下，对于一个包含 1000000 个元素的有序数组进行成功查找，折半查找比顺序查找平均快多少倍？

**折半查找：**平均情况，设所有元素等概率被查找到，有：

$$T(n) = \frac{1}{n} \sum_{i=1}^n i = \frac{1}{n} \times \frac{n(n+1)}{2} = \frac{n+1}{2}$$

**折半查找：**由分治法思想，得：  $T(n) = T(n/2) + 1$ ，  $T(1) = 1$ ，由替换法解得：

$$T(n) = \log_2 n + 1$$

代入  $n=1000000$  得  $\frac{T_{\text{顺序}}}{T_{\text{折半}}} = 25085.9$

2.N 个士兵组成的小分队必须越过一条又深又宽又没有桥的河。他们注意到在岸旁有两个 12 岁大的小男孩在玩划艇。然而船非常小，只能容纳两个男孩或者一名士兵。怎样才能让士兵渡过河并且留下两个男孩共同操纵这条船？这条船要在岸与岸之间横渡多少次？

考虑以下步骤：

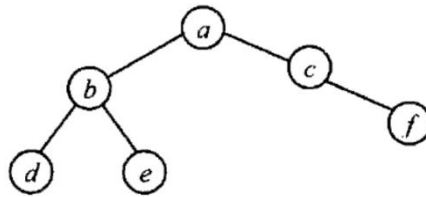
- ①两个男孩同时过河
- ②男孩 1 下船，男孩 2 返回
- ③男孩 2 下船，士兵 1 上船并过河
- ④士兵 1 下船，男孩 1 上船并过河

步骤完成后，可以帮助 1 名士兵过河，且两个男孩都回到原来位置共同操纵船，用了 4 次横渡。

由此推知，重复①-④N 次即可把 N 个士兵全部运过河，至少 4N 次横渡。

3.遍历下面的二叉树，写出遍历顺序（如：abcdef、fedcba）：

a.用前序法 b.用中序法 c.用后序法



a.abdecf

b.dbeacf

c.debfca

4.对于下面的每个列表，从一棵空树开始，通过连续插入它们的元素来构造一棵

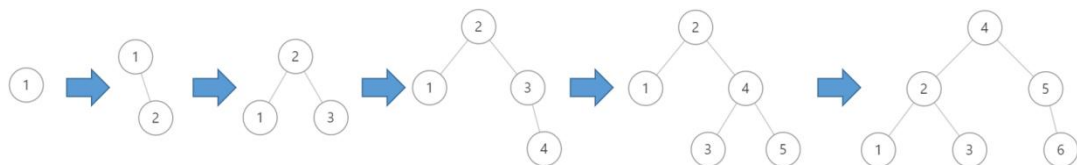
AVL 树：

a.1,2,3,4,5,6

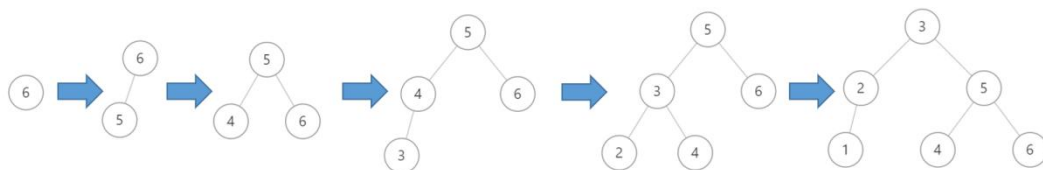
b.6,5,4,3,2,1

c.3,6,5,1,2,4

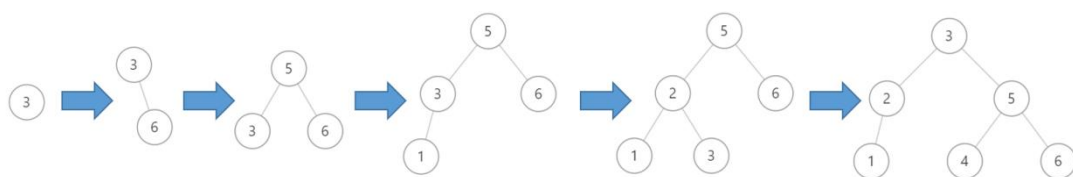
a.



b.



c.



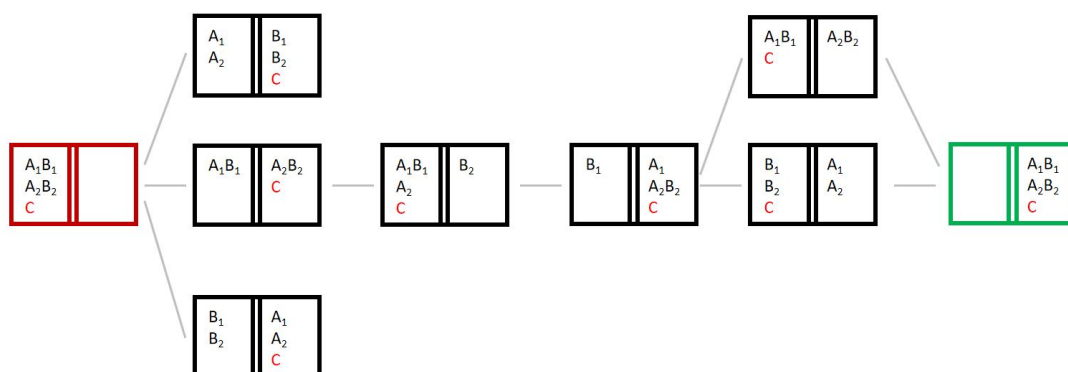
5.吃醋的丈夫谜题 有  $n$  对夫妇要越过一条河。他们有一条船，但一次最多只能载两个人。为了使情况复杂化，我们假设所有的丈夫都爱吃醋，因此在过河的全过程中，即使有他人在场，但如果没有本人的陪伴，丈夫也不会允许妻子和其他妻子的丈夫在河的另一岸上。在这种约束下，他们能越过河去吗？

- 对于  $n=2$  的情况，求过河的次数
- 对于  $n=3$  的情况，求过河的次数
- 对于任何  $n \geq 4$  的情况，这个问题有解吗？如果有，请指出他们一共要过多少次河；如果无解，请解释原因。

设船为  $C$ ， $n$  对夫妇表示为  $A_i B_i (0 < i < n)$ （丈夫为  $A$ ，妻子为  $B$ ），过河次数为  $k$ ，等价状态只画其中一种情况。

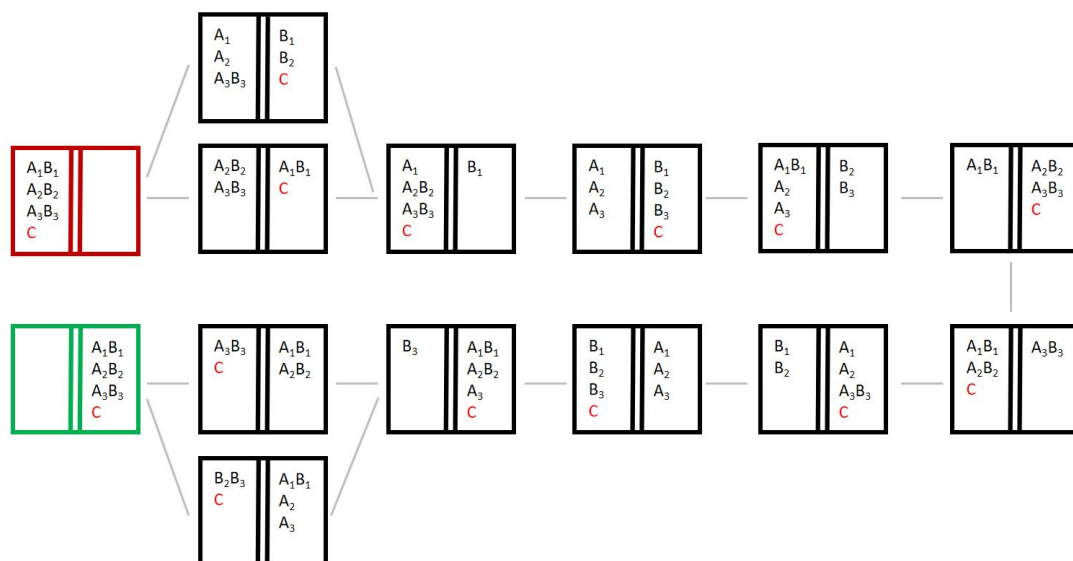
a.

$k=5$ ，解法如下：



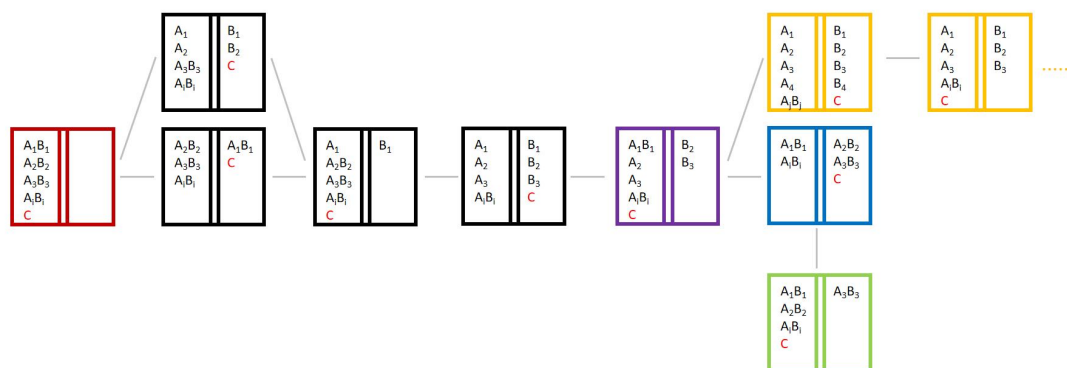
b.

k=11, 解法如下:



c.

无解, 考虑以下过程 (设  $3 < i \leq n$ ):



考虑橙色状态:

此时对岸妻子数量>2, 只要有丈夫过去 (无论 1 个丈夫、2 个丈夫、1 对夫妇), 就会因为没有办法在数量上完全匹配对岸的妻子, 而导致**发生吃醋**, 因此只能是妻子发生流动, 也因此**在橙色支线**上所有情况中, 对岸只有妻子。当且仅当回到**紫色状态**, 才有可能让任意一位丈夫过去, 此支线无法完成目标。

考虑黄绿色状态时的下一步过河安排：

- ①一夫 ( $A_1$ )：则  $B_1$  会和其余男性在原来岸上，发生吃醋
- ②一妻 ( $B_1$ )：则  $B_1$  会和  $A_3$  在对岸上，发生吃醋
- ③两夫 ( $A_1A_2$ )：则  $B_1$ 、 $B_2$  会和其余男性在原来岸上，发生吃醋
- ④两妻 ( $B_1B_2$ )：则  $B_1$ 、 $B_2$  会和  $A_3$  在对岸上，发生吃醋
- ⑤夫妻 ( $A_1B_1$ )：不发生吃醋，但回到了蓝色状态

也就是说，黄绿色状态只能回到蓝色状态，无法进行新的状态转换，故没有办法将渡河再进行下去。

综上所述， $n \geq 4$  时无解。

编写一个高效的算法来搜索  $m \times n$  矩阵 matrix 中的一个目标值 target。该矩阵具有以下特性：

- 每行的元素从左到右升序排列 ( $matrix[i][j] < matrix[i][j+1]$ )
- 每列的元素从上到下升序排列 ( $matrix[i][j] < matrix[i+1][j]$ )

若能找到，返回 true，若不能,返回 false。

**算法思想：**

由题意，既然每一行都是从左到右升序，则每一行最后一个元素一定是该行中最大的。又每一列都是从上到下升序，则最后一列一定集合了整个矩阵最大的元素。

因此，若能以此性质，利用减治法的思想减小搜索范围，则可能提高效率。

观察发现，若以搜索区域右上角的元素为基准和 target 比较：

- ①基准=target，则搜索完成

②基准>target, 说明最后一列的最小元素都要大于 target, 故最后一列所有元素都大于 target。由于最后一列有序, 可以将最后一列剔除出搜索区域。搜索区域列数-1

③基准<target, 说明第一行的最大元素都要小于 target, 故第一行所有元素都小于 target。由于第一行有序, 可以将第一行剔除出搜索区域。搜索区域行数-1

④若行数、列数有 0, 则将搜索区域缩小到 0 都没找到, 搜索失败。若行数、列数均不为 0, 则生成新的搜索区域, 以右上角元素为新基准, 重复①-③

**性能评估:**

**时间:**

**最好情况:** 原矩阵右上角元素刚好是 target, 复杂度为  $O(1)$

**平均情况:** 假设所有位置上的元素成为 target 的概率相等, 则数学期望值

$$C(m, n) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (i + j) = \frac{m+n}{2} + 1 \in \Theta(m+n)$$

**最坏情况:** 若 target 在左下角或不存在, 则每一行、每一列都要执行 1 次剔除操作,  $C(m, n) = m + n \in \Theta(m + n)$

**空间:** 需两个变量 m、n 记录行数、当前列数, 由于直观上看是从上到下依次减少行, 需再来一个变量 line 记录当前搜索区域第一行在数组中的编号, 保证在搜索区域行数为 0 时能结束循环。综上, 无论什么情况都只要 3 个变量, 空间复杂度为  $O(1)$ 。

**补充函数:**

```
bool searchMatrix(vector<vector<int> >& matrix, int target){
    int m = matrix.size(), n = matrix[0].size();
    int line = 0;

    while (m > line && n > 0) {
        if (matrix[line][n - 1] > target)
            n--;
        else if (matrix[line][n - 1] < target)
            line++;
        else
            return true;
    }

    return false;
}
```

运行结果：

---

C:\Windows\system32\cmd.exe - test

```
C:\Users\CHEMISTRYMASTER\Desktop\Assignment2>test
correct:200
error:0
用时:16ms
请按任意键继续. . .
```