

② ↓ ③ ↓
0 1 2 3 4 5 6 7 8 9 10 11 12 13

1. 二分搜索一个 14 个数的数组，查找 $A[4]$ 所经过的元素有 $A[6]$ $A[2]$ $A[4]$
2. 一个序列先入栈，再出栈，出栈元素加入队列，生成一个新的顺序（已给出），则栈结构最少需要能保存几个元素 _____。
3. 一个 5000 个元素的数据需要排序，在堆排序，基数排序，快速排序里，要求速度最快，选哪一个 基数排序
4. n 个结点的 m 序 B 树，有 $n \cdot m + 1 - n$ 个外部节点。一个 5 序 B 树有 53 个结点，该 B 树至少有 4 层。
5. 已给出一个 $K=11$ 的散列表已有三个元素，再插入两个元素，则这两个元素的位置是 _____。
6. 已给出一个无序数组，选第一个元素作为基点，快排一趟之后的顺序为 _____。
7. 一个图已给 3 条边，再添加一条边，使其有唯一的拓扑序列，添加的边是 _____，拓扑序列为 _____。
8. 已给出一个序列，初始化为最小堆 _____。

1. 跳表和散列，分别搜索最小元素 写出思想和时间复杂度。
2. 已给出一个序列，写出建立 AVL 树的过程，及删除某一个元素后的结果。
3. 已给出一个有向图，写出对应的邻接表，根据 Dijkstra 算法写出某个顶点到其余各顶点的最短路径。
4. 已给出一颗公式化描述的二叉树，画出二叉树并写出前中后序列及转化成森林。
5. 无向图用公式化描述，为简化，用数组 M 表示上三角矩阵。写出 $A[i, j]$ 到 M 的映射关系，说明如何求任意顶点 i 的度。
6. 6 个有序的序列，20 30 40 60 70 100 通过 5 次两两合并，生成一个有序的序列，求最少次数的合并过程。

九子树最大或九子树最小替换 $O(\log_2 n)$

1. 删除链表形式的 二叉搜索树的最大元素，写出思想，算法实现，时间复杂度。
2. 邻接链表表示的图 写出算法判断是否存在 $V \rightarrow U$ 的路径，以及思想。

上三角 $A[i, j]$ 行映射

第一行 n 个

第 m 行 $n - m + 1$ 个

前 $i-1$ 行共有 $\frac{(2n-i+2)(i-1)}{2}$ 个元素。

第 i 行 $j - (i-1)$ 个元素

映射到 $a[\frac{(2n-i+2)(i-1)}{2} + j - (i-1)]$

i 的度：第 i 行有 n 个元素非空度为 n

$$\frac{(n+n-m+1) \cdot m}{2} = \frac{(2n-m+1)m}{2}$$

1. 跳表和散列，分别搜索最小元素 写出思想和时间复杂度。

跳表不是有顺序的吗???

散列这个没啥规律就挨个比较每次比较把较小的那个留下来，时间复杂度 $O(n)$ 。

2. 邻接链表表示的图 写出算法判断是否存在 $V \rightarrow U$ 的路径，以及思想。

广度优先遍历。

```
int n,
void FindPath(firstNode *l, node *A, node *B){
    Queue<int> q = new Queue<int>();
    bool by[n]; for(int i=0; i<n; i++){by[i]=false;}
    q.push(A->element);
    while(!q.empty()){
        int current = q.front();
        q.pop();
        node * temp = l[current]->next;
        while(temp != null){
            if(temp->element == B->element){
                return true;
            }
            else{
                if(!by[temp->element]){
                    q.push(temp->element);
                }
            }
        }
    }
}
```

```
    }  
    return false;  
}
```