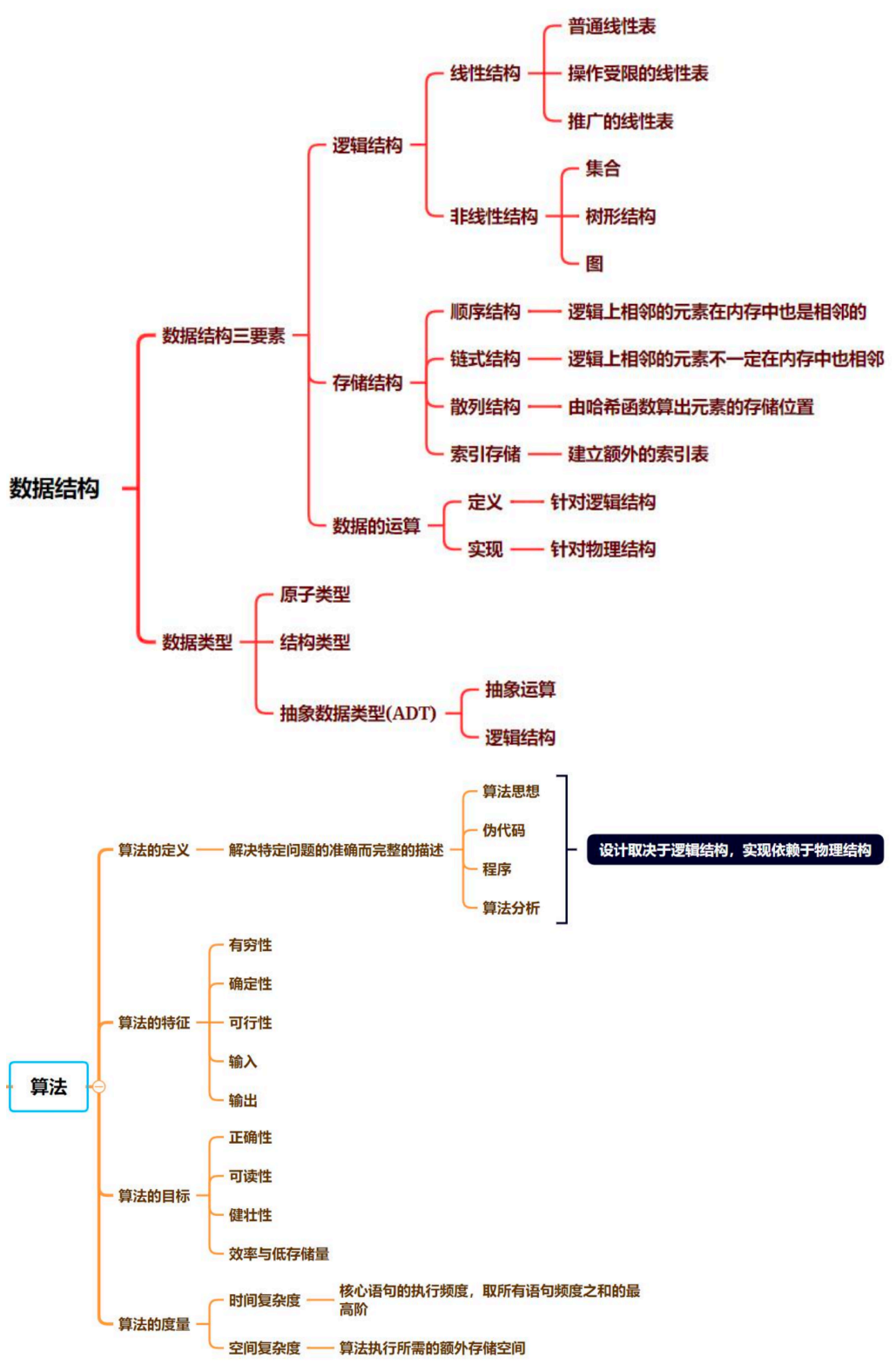


强化_1

绪论



下列函数的时间复杂度是() C

```
int foo(int n){
    int i = 0, sum = 0;
    while(sum < n){
        sum += ++i;
    }
    return i;
}
```

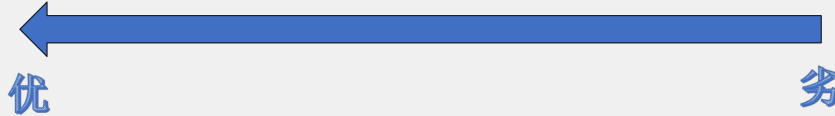
A. $O(n)$

B. $O(\log n)$

C. $O(n^{1/2})$

D. $O(n^{1/3})$

$O(1) \leq O(\log_2 n) \leq O(n) \leq O(n \log_2 n) \leq O(n^2) \leq O(n^3) \leq \dots \leq O(n^k) \leq O(2^n) \leq O(n!) \leq O(n^n)$



线性表

定义：具有相同数据类型的 n 个数据元素构成的有限序列，表中除第一个和最后一个元素外，每一个元素都有唯一的直接前驱和直接后继，第一个元素只有后继，最后一个元素只有前驱。



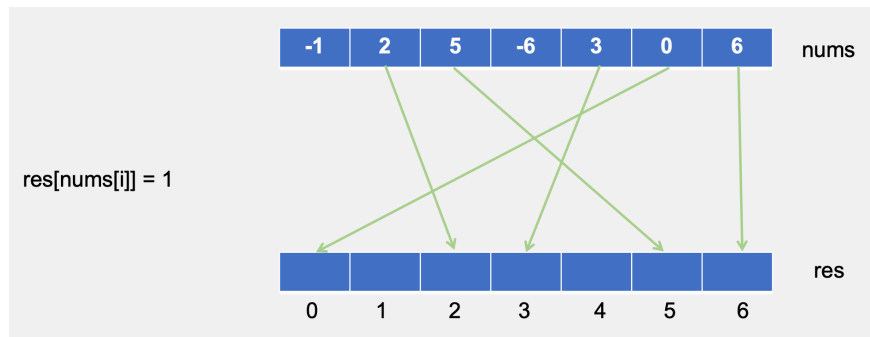
【例题】下列说法正确的是？

II

- I. 在不带头节点的单链表中，无论是插入还是删除，都需要找到其前驱；
- II. 删除带头结点的循环双链表中某节点时，只需要修改两个指针域；
- III. 将两个各有 n 和 m 个元素的递增有序表归并为一个有序表，仍保持递增有序，最少比较次数为 $m+n-1$ ；
- IV. 设单链表只有头节点，将长度为 n 的单链表接在长度为 m 的单链表后的时间复杂度为 $O(1)$ ；

【例题】现有一顺序表 L ，求出该顺序表中第一个顺序缺失的正整数，如 $\{5, 4, 7, 1, 0, -1, -3, 2\}$ 中第一个顺序缺失的正整数是3，而 $\{1, -1, -3, 0, 4\}$ 中第一个顺序缺失的正整数是2(表中元素的最大值不超过表长的值)。

- (1) 写出时间上最优的算法思想；
- (2) 用 c 或 $c++$ 实现代码，关键语句给出注释；
- (3) 写出算法的时间复杂度和空间复杂度；



```
int firstMissingNum(SqList& L) {  
    int n=L.length;  
    int nums[]=(int*)malloc(sizeof(int)*(n+1));  
    int i=0;  
    while(i<=n) {    // 初始化数组  
        nums[i++]=0  
    }  
    for(i=0; i<n; i++) {    // 建立映射  
        if(L.data[i]>0) {  
            nums[L.data[i]]=1;  
        }  
    }  
    for(i=1; i<=n; i++) {    // 寻找缺失的第一个正整数  
        if(nums[i]==0) break;  
    }  
    return i;  
}
```

栈和队列

数组、矩阵和串