

2020 真题答案

一、单项选择题

1. B

2. D 解析：算法分析主要以时间复杂度分析为主，所以一般是时间复杂最差情况下的空间复杂度

3. D 解析：根据书上的表示（树这部分的意思：第一点根节点无前驱；第二点根节点对应的子树的结点没有公共结点，只有一个结点与根节点相连；第三点没有相同的边），答案在树和图中间选择，再加上题目说最可能

ADT Tree {

 数据对象 D:D 是具有相同特性的数据元素的集合。

 数据关系 R:若 D 为空集，则称为空树；

 若 D 仅含一个数据元素，则 R 为空集，否则 R = {H}, H 是如下二元关系：

- (1) 在 D 中存在唯一的称为根的数据元素 root, 它在关系 H 下无前驱；
- (2) 若 $D - \{root\} \neq \emptyset$, 则存在 $D - \{root\}$ 的一个划分 $D_1, D_2, \dots, D_m (m > 0)$, 对任意 $j \neq k (1 \leq j, k \leq m)$ 有 $D_j \cap D_k = \emptyset$, 且对任意的 $i (1 \leq i \leq m)$, 唯一存在数据元素 $x_i \in D_i$, 有 $\langle root, x_i \rangle \in H$;
- (3) 对于 $D - \{root\}$ 的划分, $H - \{\langle root, x_1 \rangle, \dots, \langle root, x_m \rangle\}$ 有唯一的一个划分 $H_1, H_2, \dots, H_m (m > 0)$, 对任意 $j \neq k (1 \leq j, k \leq m)$ 有 $H_j \cap H_k = \emptyset$, 且对任意 $i (1 \leq i \leq m)$, H_i 是 D_i 上的二元关系, $(D_i, \{H_i\})$ 是一棵符合本定义的树, 称为根 root 的子树。

其余结点可分为 $m (m > 0)$ 个互不相交的有

限集 T_1, T_2, \dots, T_m , 其中每一个集合本身又是一棵树，并且称为根的子树(SubTree)。例如，在图 6.1 中，(a) 是只有一个根结点的树；(b) 是有 13 个结点的树，其中 A 是根，其余结点分成 3 个互不相交的子集： $T_1 = \{B, E\}$, $F, K, L\}$, $T_2 = \{C, G\}$, $T_3 = \{D, H, I, J, M\}$; T_1, T_2 和 T_3 都是根 A 的子树，且本身也是一棵树。例如 T_1 , 其根为 B, 其余结点分为两个互不相交的子集; $T_{11} = \{E, K, L\}$, $T_{12} = \{F\}$ 。 T_{11} 和 T_{12} 都是 B 的子树。而 T_{11} 中 E 是根, $\{K\}$ 和 $\{L\}$ 是 E 的两棵互不相交的子树, 其本身又是只有一个根结点的树。

上述树的结构定义加上树的一组基本操作就构成了抽象数据类型树的定义。

数据关系 R:

$$R = \{VR\}$$

$VR = \{\langle v, w \rangle \mid v, w \in V \text{ 且 } P(v, w), \langle v, w \rangle \text{ 表示从 } v \text{ 到 } w \text{ 的弧,}$
谓词 $P(v, w)$ 定义了弧 $\langle v, w \rangle$ 的意义或信息 }

4. C

5. A

6. C

7. D

8. A

9. D

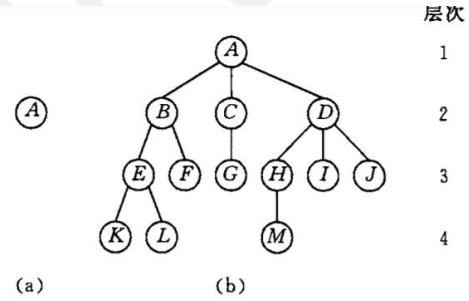


图 6.1 树的示例

(a) 只有根结点的树; (b) 一般的树

10. A
11. B
12. C
13. B 解析: 设运行次数为 k, 即 $1+2+3+\dots+k < n$, 可得 k 约等于 $n^{0.5}$
14. C
15. A
16. D 解析: 等价于 1101 0101 异或 0100 1010
17. C
18. A
19. B
20. A 解析: 012 为八进制, 0x12 为 16 进制
21. B 解析: 等价于输出 s[4]
22. C
23. A
24. C
25. B 解析: 这里的 S 是 struct M 类型的结构体的别名, 不是变量, 如果没有 typedef 才是变量, 且不能用 M 来定义变量 (部分编译器不会报编译错误, 正常运行), 只能用 struct M 或者 S 来定义结构体变量
26. D

二、填空题

1. 链式
2. /* a + b c 2
3. 99
4. 3
5. 25 解析: 计算公式 $1+2+3+4+5+6+5-1 = 25$
6. 2e
7. 2n-1
8. 递增 (意思相近也可)
9. NULL 解析: 唯一的结点被删除, 此时链表为空
10. q->next!=p 解析: 查找 p 结点的前驱结点
11. p->data 解析: 题目存在问题应当是 q->sum -= p->data; 下面一句也是
12. p->next 解析: 进行删除操作
13. 0
14. >
15. m+1 解析: 左右子树最高的高度加 1
16. 循环结构
17. 0
18. (i-5)%9==0
19. strlen(a)
20. k=i 解析: 修改最大元素标记
21. k<len-1 解析: 不为最后一个元素, 意思相近的代码也可, 如 k!=len-1
22. m=0 解析: 初始化
23. a[m] = x 或 a[i]=x 解析: 插入

24. `insert(a, x)` 解析: 调用函数
25. `s->data < p->data` 或者 `p->data > s-data` 解析: 查找插入位置
26. `p->next`
27. `q->next = s`
28. `a%16`
29. `a>0` 或 `a!=0`
30. `p>=s`

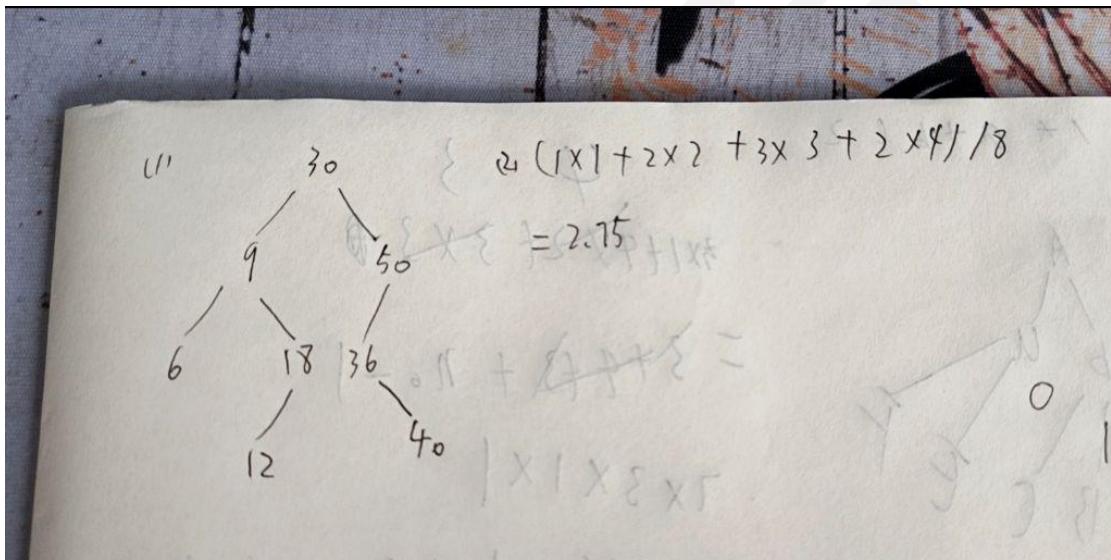
三、解析题

1.

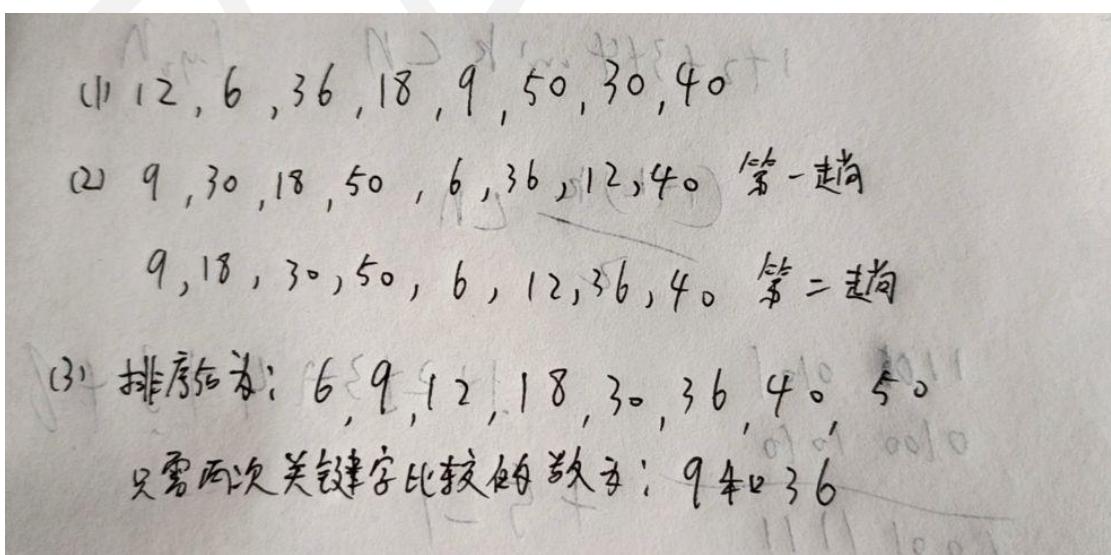
(1) 1321 解析: 最外层输出 21, 然后往里依次是 3, 1, 即 1321

(2) $\lfloor \log_2 n \rfloor + 2$

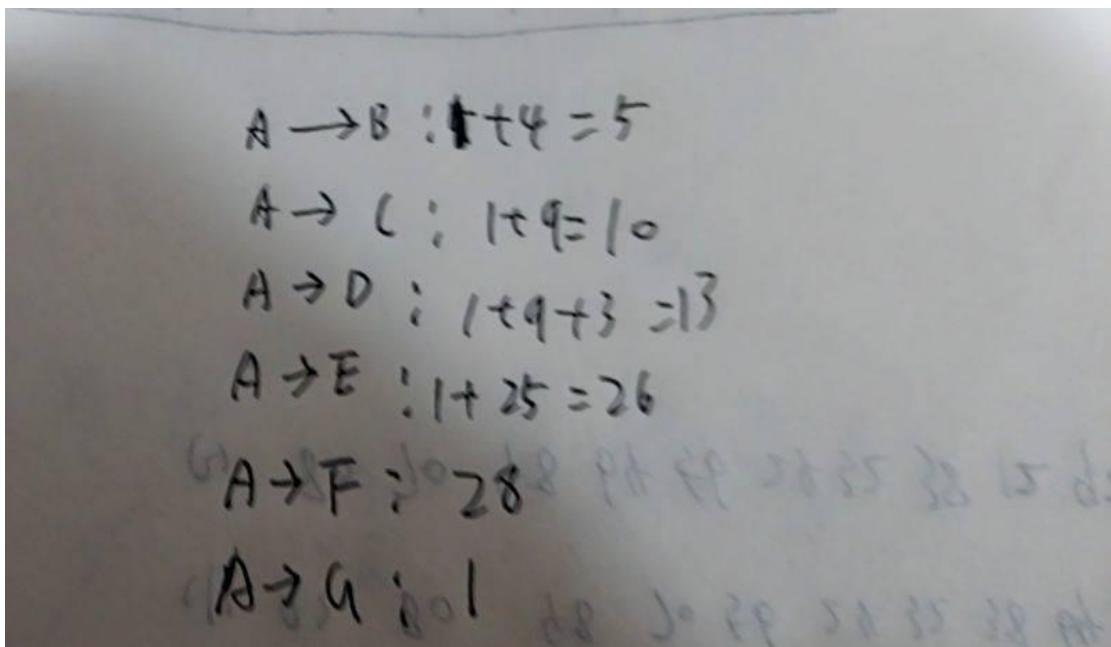
2.



3.



4.



5. 245
6. bcdefa
7. 9 19 解析:注意 m 和 i 都是静态变量
8. 4321dcba
9. 2,4
10. 73, 82

四、编程题

1.

```
int count( BT bt, int x)
{
    int sum = 0;
    if(bt == NULL) return 0;
    if(bt->data < x) sum+= bt->data;      //判断当前结点
    sum += count(bt->lchild, x);           //计算左子树
    sum += count(bt->rchild, x);           //计算右子树
    return sum;
}
```

2.

```
#include <stdio.h>
#define N 10
double fun(double *p, int n){
    int min, max, i, sum;
    sum = min = max = p[0];
    for(i = 1; i < n; i++){
        if(p[i] < min) min = p[i]; //找出最小值
        if(p[i] > max) max = p[i]; //找出最大值
        sum += p[i]; //求和
    }
}
```

```
    }
    sum = (sum - min -max) / (n-2); //求平均成绩
    return sum;
}
int main()
{
    double p[N], avg;
    int i;
    for(i=0;i<N;i++){
        scanf("%lf", &p[i]);
    }
    avg = fun(p, N);
    printf("%.2lf", avg);
    return 0;
}
```