

2021 真题答案

一、单项选择题

1. B 解析：算法是指解决方案的准确而完整的描述，即问题求解的步骤

2. D 解析：循环队列是顺序表

 哈希表对应哈希存储

 线索树：在链式存储结构的基础上对树进行线索

 栈不能体现是顺序存储还是链式存储

3. B

4. D

5. A

6. C 解析：插入修改 rear 指针，删除修改 front 指针，且基本都是加值

7. D

8. C

9. C

10. A

11. D

12. A

13. B

14. C

15. A

16. D

17. C 解析：共用体所有成员共享一个内存，每次只能存放其中一个成员，且成员可以为结构体类型

18. A 解析：没有'\0'的字符数组不是字符串，同时 '\0' 也需要一个空间的内存

19. B

20. A

21. B 解析：12 赋值给了 x，345 赋值给了 y，同时输入缓存区还存在一个数字 6，输出会正常输出

22. D

23. C

24. D

25. B

26. A

二、填空题

1. 操作

2. 顺序

3. $Ab+c/d-e^*$

4. 211 解析：完全二叉树可能无度为 1 的结点，可能有一个度为 1 的结点（但至多只有一个），度为 2 的结点数比度为 0 的结点数少 1，所以该树度为 0 的结点为 211，无度为 1 的结点

5. 19 解析： $(1+2+3+4+5+6+6) - 1 = 26$

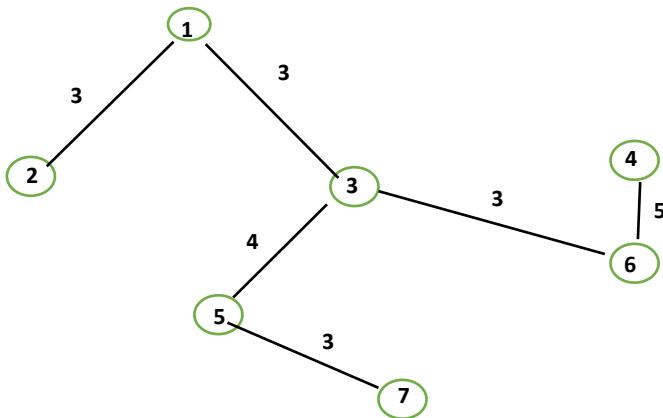
6. n

7. 邻接表

8. $2n-1$
9. $p \rightarrow \text{data} == x$
10. $>$
11. $p \rightarrow \text{next}$
12. p
13. $\text{bt} \rightarrow \text{lchild}$
14. 1 解析: 没有左右孩子, 只有当前一个结点
15. $\text{bt} \rightarrow \text{lchild} \rightarrow \text{num} + \text{bt} \rightarrow \text{rchild} \rightarrow \text{num} + 1$
16. void
17. 13
18. $i++$ 解析: 继续在 i 后面赋值, $i+j$ 答案错误, 因为 $j++$ 会使 j 的值多 1
19. $j++$ 解析: j 从 0 开始扫描赋值
20. wb 解析: `fwrite` 为二进制写, 所以需要以二级制形式打开
21. `use.code[j] -= 1`
22. `fclose(fp)`
23. $j < i$
24. $a[i][j] = a[i-1][j] + a[i-1][j-1]$ 解析: 杨辉三角计算公式
25. $j \leq i$
26. $d=1.0$
27. $k \leq n$
28. $p = \text{first} \rightarrow \text{next}$ 解析: m 初始为第一个结点的数据域值
29. $p = p \rightarrow \text{next}$
30. $m = p \rightarrow \text{data}$

三、j 简答题

- 1.
- (1) 1 2 3 5 7 4 6
- (2) 是 1 2 4 6 3 5 7 解析: 答案不唯一
- (3)



2.

0	1	2	3	4	5	6	7	8	9	10
22	1	13	12		38	27		30	41	

$$ASL = (1+1+1+3+1+1+2+2)/8 = 1.5$$

3.

(1)12,20,15,6,8,30,45,72,63,54

(2)6,8,12,45,20,15,54,72,30,63

(3)72,63,30,45,54,15,6,20,12,8

4.: 此题答案省略了活动的最早开始时间和最晚开始时间, 考试时建议补全

	A	B	C	D	E	F	G
Early	0	1	2	3	6	10	14
Late	0	4	2	5	6	10	14

关键路径: ACEFG

5. 3,3,2,4,30,1

6. ar=4 ar=4 ar=5 解析: 这里存在经典的 `printf("%d", *(p++))` 模式, 注意++特性(无论有没有括号都不影响它参与其他运算), 且 c 语言在编译时字符、字符串的处理会优先于宏替换, 所以双引号内部的 ar 没有替换

7. 5 6 7 8 9 10 3 1 4 2

8. 11 20 201

9. t**M

10. 9 2 7 4 5 6 3 8 1 10

解析: fun 函数中 i 的取值为 0,2,4,6,8, a 初始化值为 1,2,3,4,5,6,7,8,9

当 a[i] 小于 a[j] 的时候进行交换, 且是多个 j 与变化后的 i 进行交换

当 i 为 0 时, j 的取值为 2,4,6,8 得 a 数组为 9 2 1 4 3 6 5 8 7 10

当 i 为 2 时, j 的取值为 4,6,8 得 a 数组为 9 2 7 4 1 6 3 8 5 10

当 i 为 4 时, j 的取值为 6,8 得 a 数组为 9 2 7 4 5 6 1 8 3 10

当 i 为 6 时, j 的取值为 8 得 a 数组为 9 2 7 4 5 6 3 8 1 10

当 i 为 8 时, j 的取值无得 a 数组为 9 2 7 4 5 6 3 8 1 10 (无交换)

四、编程题

1.

```
void fmin(BT boot, int *pmin)
{
    if(bt==NULL) return;
    if(bt->data < *pmin) *pmin = bt->data;
    fmin(bt->lchild);
    fmin(bt->rchild);
}
```

2.

```
#include <stdio.h>
void input(int number[], int n)
{
```

```
int i;
for(i=0; i<n; ++i){
    if(i==n-1)
        scanf("%d", &number[i]);
    else
        scanf("%d,", &number[i]);
}
}

void max_min_change(int number[ ], int n)
{
    int i,max = number[0], index_max= 0, temp, min = number[0], index_min ;
    for(i = 1;i<n;++i){
        if(number[i]>max){
            max = number[i];
            index_max = i
        }
        if(number[i]<min){
            min = number[i];
            index_min = i
        }
    }
    temp = number[0];
    number[0] = number[index_max];
    number[index_max] = temp;
    temp = number[9];
    number[9] = number[index_min];
    number[index_min] = temp;
}
void output(int number[ ], int n)
{
    int i;
    for(i=0; i<n; ++i){
        if(i==n-1)
            printf("%d", number[i]);
        else
            printf("%d,", number[i]);
    }
}

void main(){
    int number[10];
    input(number, 10);
    max_min_change(number, 10);
    output(number, 10);
```

文彦考研西南交大计算机 840 交流群：574690356
西南交大研究生白竹学长在线答疑，不定期开设讲座，欢迎大家加入

}

