数据结构答案整理与解析

选择：

1. B
2. D

算法（Algorithm）是指解题方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。

1. B

存储及运算都需考虑数据元素本身的形式、内容等。而逻辑结构中关心元素之间的逻辑关系，与数据元素本身无关。

1. C

动态、静态 -》 链表、数组

顺序、链式 -》 数组、链表

初等 -》 线性

构造型 -》构造产生的类型

1. D

A：存储可以是顺序 / 链式

B：逻辑是脑子里面想的，存储是计算机中实际的

逻辑分：线、树、网。。

存储：顺序、链式、、

C:存储会影响效率，比较数组与链表

1. C

书本概念。。。

1. B

自己设定一个n，套一下

1. D

A:单链表需要遍历

C:双链表需要遍历

B:带一个结点指针也需要遍历

1. A

尾指针的next指向头指针是循环链表特点

1. A

自己设定一个i，套一下，不是从0开始数的

1. B

第一次比较下标为 5 的，第二次比较下标为8的

12、B

2836-2100 = 736 736/4 = 184 184个元素 = 54\*3 + 22 54\*3=>R[3][0][0]

22= 18+4 => [3][4] => R[3][3][4]

13、B

Malloc可能申请到连续的地址，所有可连续也可不连续

1. D
2. B
3. D

删除一个元素之后，后续结点移动n-i下

1. D

（1+2+3+。。+n）/n => (0.5\*n^2 + 0.5n)/n => n+1/2

1. A
2. D 重复题
3. A
4. D
5. D答案存在争议，强行解释的话，就是B选项是后++的
6. D

队尾指向插入位置,从11计数

队头20 - 25 =》 6

队尾11 - 0 =》 12

1. B
2. B

因为函数调用需要消耗时间

1. C

n是最先出栈的，所以必定是最后一个进栈的，它前面的数值此时全部进栈，且都不没栈

1. C

Dce没问题，ab不可能，只能是ba

1. D

1入栈 2、3入栈出栈 4入栈、5入栈出栈， 此时栈顶为4

1. B
2. C

基本概念

1. B
2. D

为了防止下标越界 %m

1. A
2. A

队头f指向出队元素，队尾r指向插入位置

假设m = 10

F = 1，r = 4 队中元素为3个 下标为1 2 3

R = 2，f = 6 队中元素为 6个 下标为 6 7 8 9 0 1

带入4个式子 (f-r + m)%m 不成立 (r-f+m)%m 成立

34、A

列优先： 行列转换看

A[12][18] -> 18行12列

A[0][0] -> 150

A[9][7] => 7\*12 + 9 => 84 + 9 => 93 每个元素3个字节 =》 93\*3 =》279 + 150 = 429

35、C

自己画个根结点，度是有几个孩子

1. C

空域：结点的孩子个数不为两个时 -》 空着的孩子表示空域

37、B

假设序列为

先：ABC

后：CBA

其树推出 -》

A A

B B

C C

排除ACD

38、A

度0结点树 ：4

度1结点数： x

度2结点数： y

结点数：n = 10

树的分支方程： n-1 = x + 2y

树的结点方程： n = x+y+4

解方程： -1 = y-4 => y = 3 ,x = 3

1. C
2. A

画一个三个结点的树，测试下

1. B

根据孩子兄弟转化方式，转化，右边结点是兄弟，根节点不可以有兄弟，

否则是森林

1. B

除第一层外，其它层至少两个结点

1. C

度为1的结点不存在

1. A

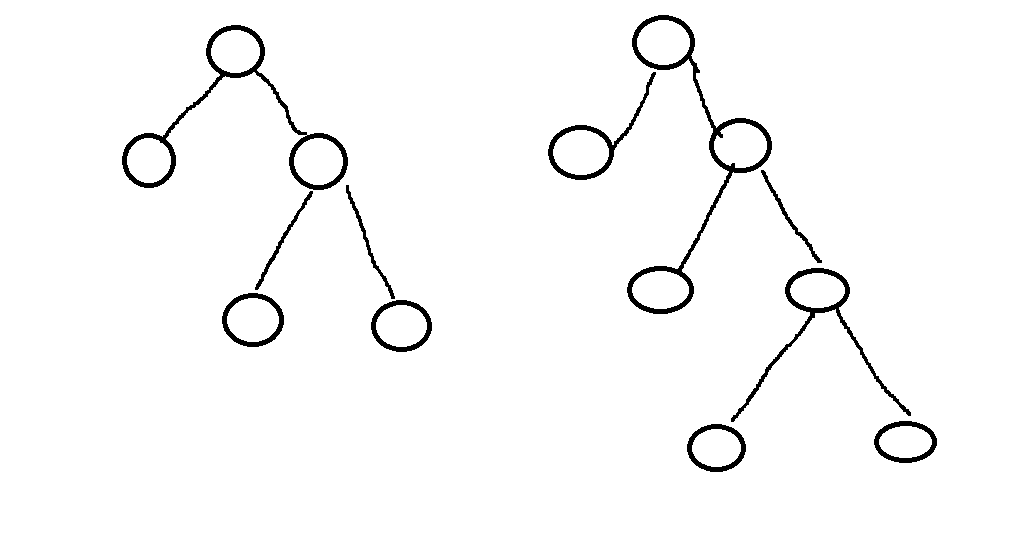
左孩子 n\*2 右孩子N\*2 +1

1. C
2. A
3. A

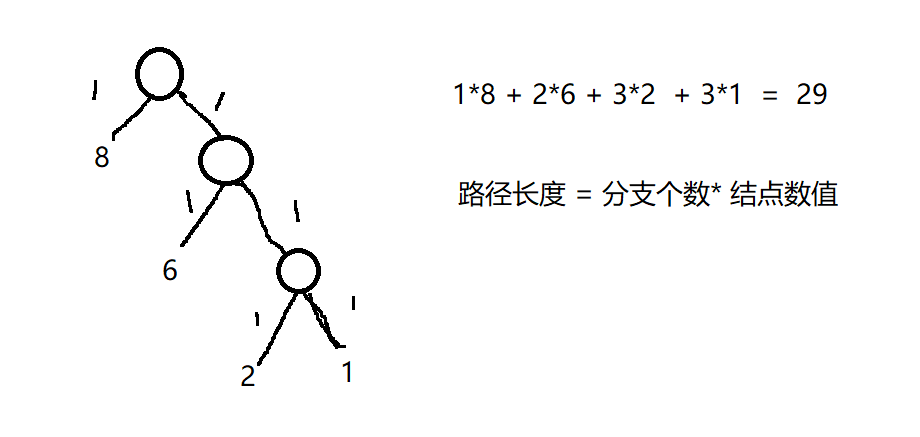
总是先遍历坐标的叶子节点再遍历右边的叶子结点

1. D

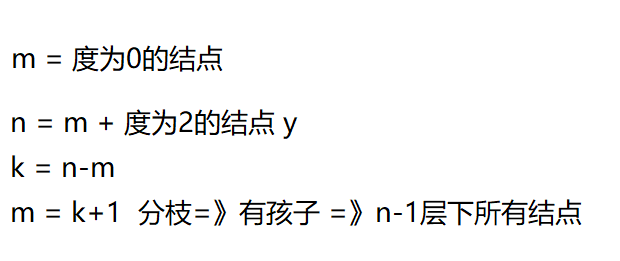
哈弗曼树的形状总是如下的：



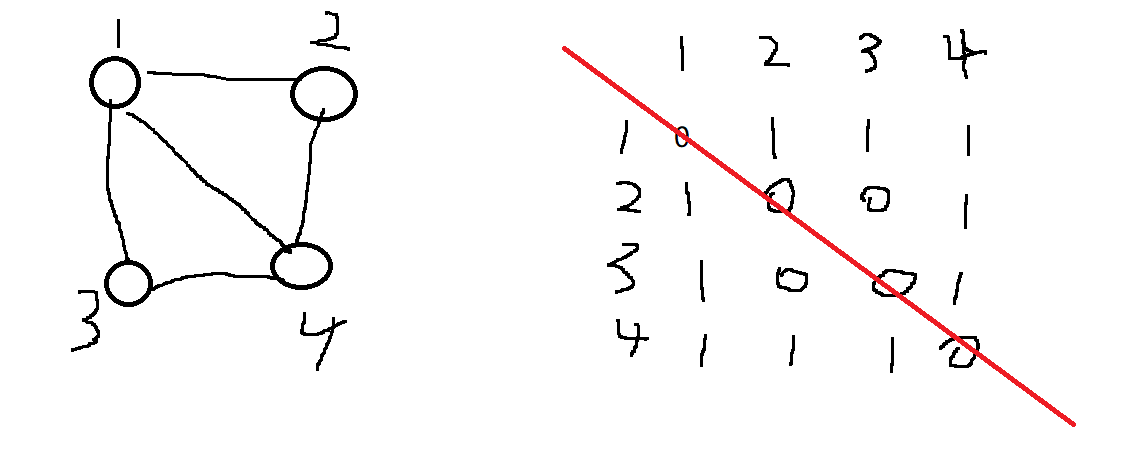
1. D



1. D

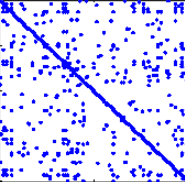


1. D

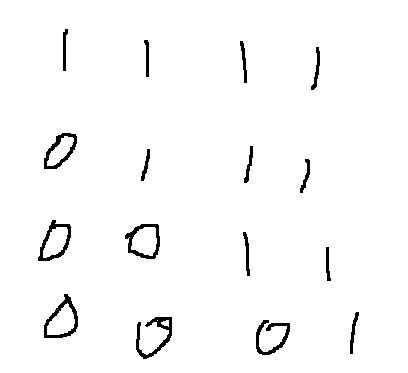


对角矩阵：除了那根红线，其他的都是0

稀疏矩阵：和二维码差不多



上三角矩阵：



1. C

自己画个图

1. B

自己画个图

1. B

深度优先：会使得树的层次较多

广度优先：单层结点多

Prim ：类似于迪杰斯特拉

拓扑排序：<https://blog.csdn.net/qq_38984851/article/details/82844186>

1. A

两个顶点的图

连通图：任意两个顶点都有关系的图

A-B

56、D

完全有向图：任意两个顶点都有两个相互的弧

A《=》B

57、A

58、D

59、B

60、D

61、A

62、A

<https://blog.csdn.net/cbwem/article/details/78692408>

1. C 重复题
2. B
3. A

边 -》 无向的

66、B 重复题

67、28

7+6+5+4+3+2+1

1. C
2. C
3. B
4. A
5. C
6. D
7. B
8. C
9. D

79、A

80、A

81、D

82、B

最短路径可以有多条，所以生成树，可以有多个

1. D

比较多少次就在多少层

1. B
2. B
3. A
4. B
5. D
6. D
7. A 或者 C

A为现代的更新版

C为老版 -》从最后一个来

1. C
2. C
3. D
4. D C
5. C
6. D
7. A 或 B 根据具体排列序列来定
8. D
9. C
10. A 或者 C 重复
11. D

101、B

102、C

103、B

104、C

快排在不使用递归的情况下，不会使用额外内存

1. C
2. C
3. D
4. D