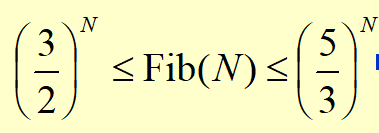
1. The Fibonacci number sequence {*FN*​} is defined as: *F*0​=0, *F*1​=1, *FN*​=*FN*−1​+*FN*−2​, *N*=2, 3, .... The time complexity of the function which calculates *FN*​ recursively is Θ(*N*!).

F The time complexity is O(2^N)

1. O(N^2) is the same as O(1+2+3+⋯+N).

T

1. N^0.01 is O(logn)

F n的多少次最后都比logn大

1. For the following piece of code the lowest upper bound of the time complexity is O(N^3).

if ( A > B ){

for ( i=0; i<N\*2; i++ )

for ( j=N\*N; j>i; j-- )

C += A;

}

else {

for ( i=0; i<N\*N/100; i++ )

for ( j=N; j>i; j-- )

for ( k=0; k<N\*3; k++)

C += B;

}

T else后的语句当N>100时，i>N就不执行第二个分循环了，故最低上限是O(N^3)

1. The recurrent equations for the time complexities of programs P1 and P2 are:

P1: T(1)=1,T(N)=T(N/3)+1

P2: T(1)=1,T(N)=3T(N/3)+1

Then the correct conclusion about their time complexities is:

A.they are both O(logN)

B.O(logN) for P1, O(N) for P2

C.they are both O(N)

D.O(logN) for P1, O(NlogN) for P2

1. P1: T(N)=T(N/3)+1=T(N/9)+2=...

P2: T(N)=3T(N/3)+1=3[3T(N/9)+1]+1=3[3(3T(N/27)+1)+1]+1

3\*1+1 3\*(3\*1+1)+1

1. To delete p from a doubly linked list, we must do:

p->prior->next=p->next; p->next->prior=p->prior;

1. To merge two singly linked ascending lists, both with N nodes, into one singly linked ascending list, the minimum possible number of comparisons is:

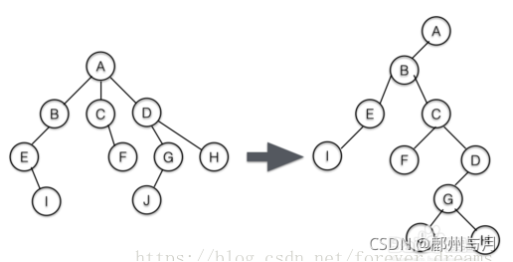
N

1. Push 5 characters ooops onto a stack. In how many different ways that we can pop these characters and still obtain ooops?

5 这题可以看做O1 O2 O3 P S五个字母的push、pop操作

1. 在一棵度为4的树T中，若有20个度为4的节点，10个度为3的节点，1个度为2的节点，10个度为1的节点，则树T的叶节点个数 ()

解答： 1.sum(所有结点个数)=20x4+10x3+1x2+10x1+1(根结点)=123个结点 2.sum(叶子节点个数)=123-20-10-1-10=82

1. If a general tree T is converted into a binary tree BT, then which of the following BT traversals gives the same [sequence](https://so.csdn.net/so/search?q=sequence&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/best_LY/article/details/_blank) as that of the post-order traversal of T?

T：

前序preorder：ABEICFDGJH

中序无（多节点没法定义中序）

后序：IEBFCJGHDA

BT：

前序preorder: ABEICFDGJH

中序inorder：IEBFCJGHDA

后序postorder:IEBFJHGDCBA

层级levelorder：ABECIFDGJH 答案：inorder traversal

1. There exists a binary tree with 2016 nodes in total, and with 16 nodes having only one child.

F 假设没有孩子的结点（叶结点）个数为n₀，只有一个孩子的结点（度为1的结点）个数为n₁，有两个孩子的结点（度为2的结点）个数为n₂。 则n₀+n₁+n₂=2016 ∵n₀=n₂+1（二叉树的性质：叶结点个数等于度为2的结点个数加1） ∴n₀+n₁+n₂=2016 ⇨n₂+1+16+n₂=2016 ⇨2n₂=1999 n₂除不尽，所以答案错误。

1. In a binary search tree, the keys on the same level from left to right must be in sorted (non-decreasing) order.

T

1. In a binary search tree which contains several integer keys including 4, 5, and 6, if 4 and 6 are on the same level, then 5 must be their parent.

F(like 5 3 7 2 4 6 8)

1. Given a binary search tree with its preorder traversal sequence { 8, 2, 15, 10, 12, 21 }. If 8 is deleted from the tree, which one of the following statements is FALSE?

A.One possible preprder traversal sequence of the resulting tree may be { 2, 15, 10, 12, 21 }

B.One possible preprder traversal sequence of the resulting tree may be { 10, 2, 15, 12, 21 }

C.One possible preprder traversal sequence of the resulting tree may be { 15, 10, 2, 12, 21 }

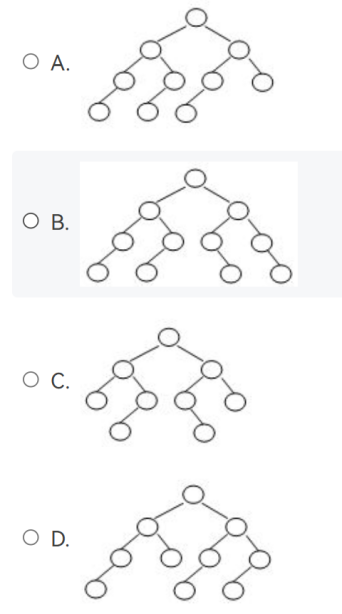
D.It is possible that the new root may have 2 children

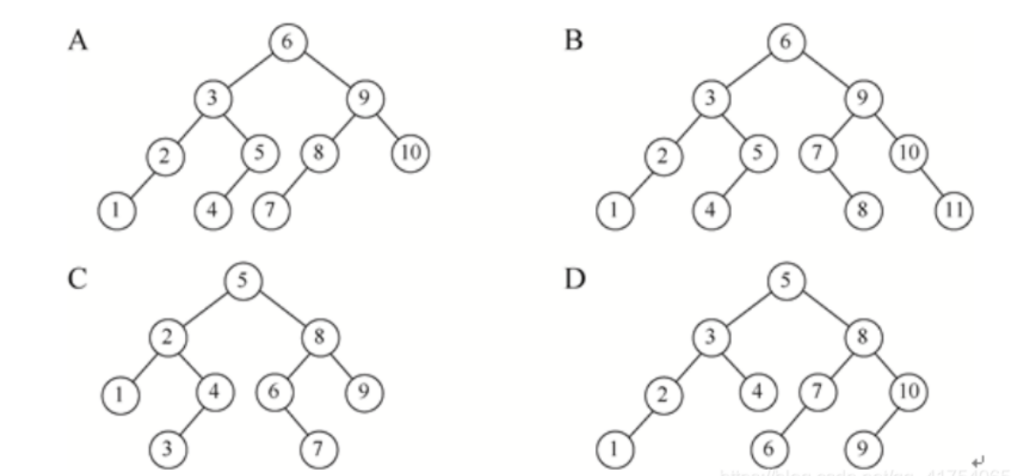
C

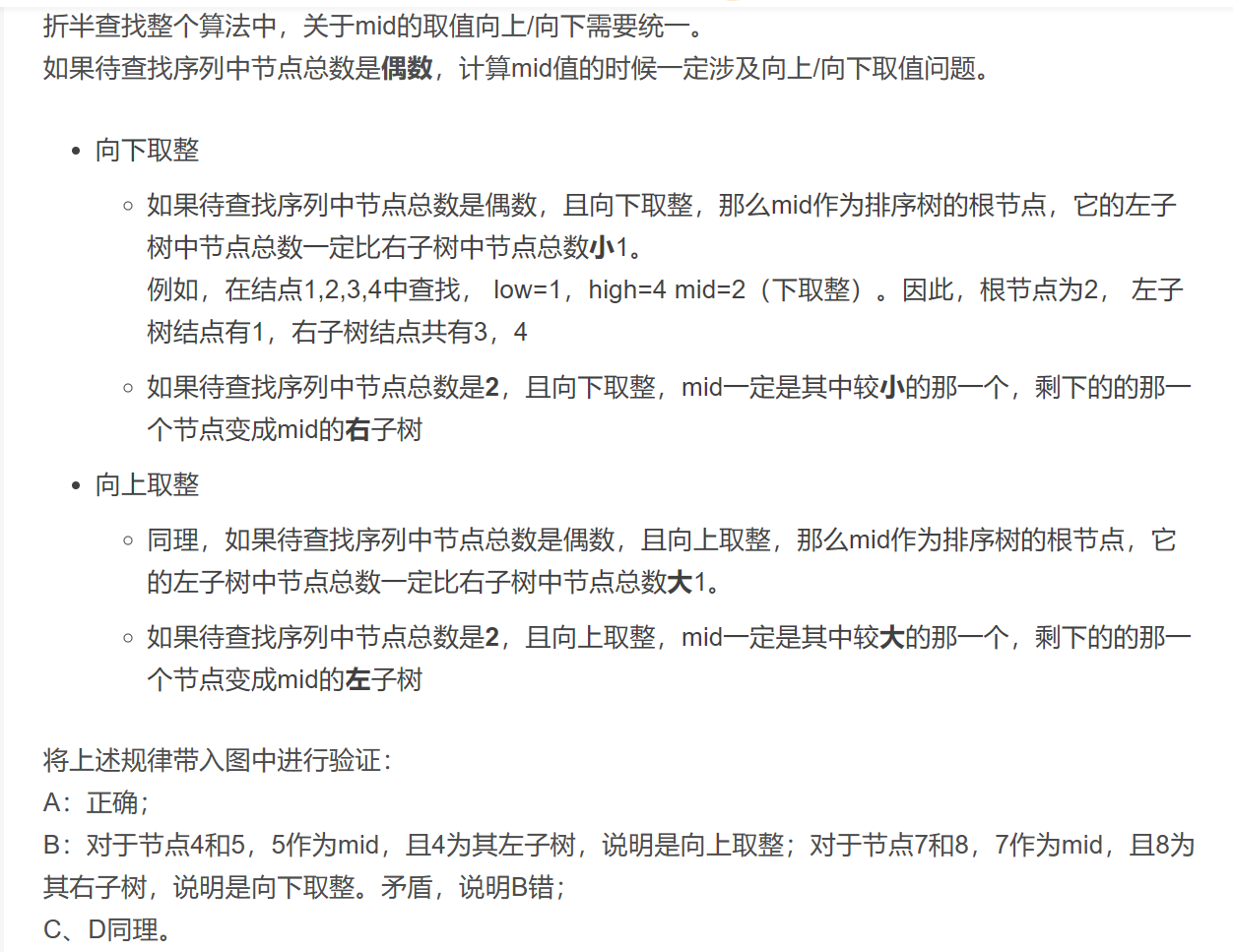
1. Insert {5, 2, 7, 3, 4, 1, 6} one by one into an initially empty binary search tree. The postorder traversal sequence of the resulting tree is: 1, 4, 3, 2, 6, 7, 5
2. Among the following binary trees, which one can possibly be the decision tree (the external nodes are excluded) for binary search?

A

解析：[折半查找](https://so.csdn.net/so/search?q=%E6%8A%98%E5%8D%8A%E6%9F%A5%E6%89%BE&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qq_41754065/article/details/_blank)判定树实际上是一棵二叉排序树，它的中序序列是一个有序序列。可以在树结点上依次填上相应的元素，符合折半查找规则的树即是所求。







1. In a max-heap with n (>1) elements, the array index of the minimum key may be \_\_.
2. 1 B. ⌊n/2⌋−1 C. ⌊n/2⌋ D. ⌊n/2⌋+2
3. Using the linear algorithm to build a min-heap from the sequence {15, 26, 32, 8, 7, 20, 12, 13, 5, 19}, and then insert 6. Which one of the following statements is FALSE?
4. The root is 5
5. The path from the root to 26 is {5, 6, 8, 26}
6. 32 is the left child of 12
7. 7 is the parent of 19 and 15

·build a min-heap的过程是 先建一个数组存放堆（先形成一个完全二叉树），再进行重排序(调整树的顺序)C

·如果题目是insert one by one，则不是建一个数组存放堆，而是直接insert的方式插入一个数就直接排序

1. If a d-heap is stored as an array, for an entry located in position i, the parent, the first child and the last child are at:

⌊(i+d−2)/d⌋,  (i−1)d+2,  id+1

1. If a binary search tree of N nodes is complete, which one of the following statements is FALSE?
2. the average search time for all nodes is O(logN)
3. the minimum key must be at a leaf node
4. the maximum key must be at a leaf node
5. the median node must either be the root or in the left subtree

C

1. In Union/Find algorithm, if Unions are done by size, the depth of any node must be no more than N/2, but not O(logN)

F 解析：Let T be a tree created by union-by-size with N nodes, then the height of T can be

at most log2(N)+1

1. The array representation of a disjoint set containing numbers 0 to 8 is given by { 1, -4, 1, 1, -3, 4, 4, 8, -2 }. Then to union the two sets which contain 6 and 8 (with union-by-size), the index of the resulting root and the value stored at the root are:
2. 1 and -6
3. 4 and -5
4. 8 and -5
5. 8 and -6

B

1. If a directed graph G=(V, E) is weakly connected, there must be at least |V| edges in G.

F

at least |V-1| in G

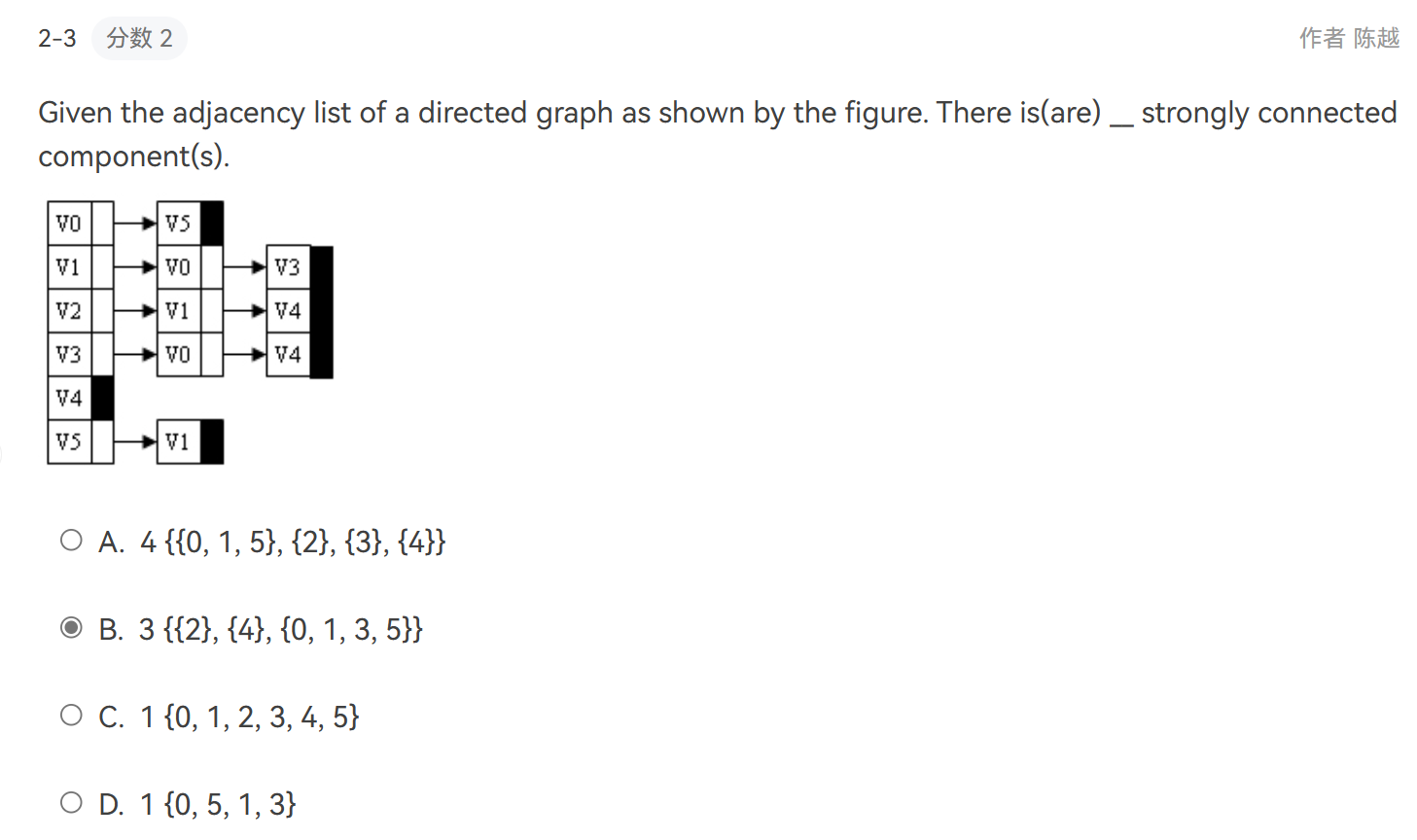
将[有向图](https://baike.baidu.com/item/%E6%9C%89%E5%90%91%E5%9B%BE?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E5%BC%B1%E8%BF%9E%E9%80%9A%E5%9B%BE/_blank)的所有的有向边替换为无向边，所得到的图称为原图的基图。如果一个[有向图](https://baike.baidu.com/item/%E6%9C%89%E5%90%91%E5%9B%BE?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E5%BC%B1%E8%BF%9E%E9%80%9A%E5%9B%BE/_blank)的基图是连通图，则有向图是[弱连通图](https://baike.baidu.com/item/%E5%BC%B1%E8%BF%9E%E9%80%9A%E5%9B%BE?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E5%BC%B1%E8%BF%9E%E9%80%9A%E5%9B%BE/_blank)

1. If graph G is NOT connected and has 35 edges, then it must have at least \_\_\_\_ vertices.

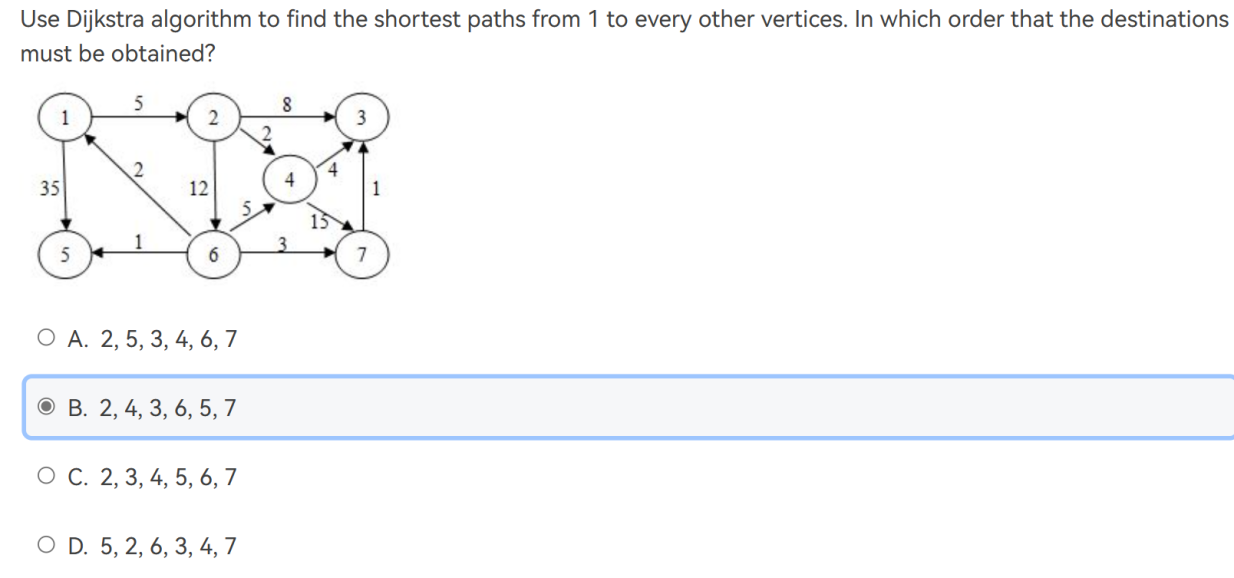
if a graph is not connected, it can have at most, the rest n-1 verticies are all connected, C(n-1,2) edges. n-1\*n-2/2>35, n=10

1. A graph with 90 vertices and 20 edges must have at least \_\_ connected component(s).

Least is 70，20个有边 50个没边；most is 84（20<6\*7/2, 90-7+1=84）

1. 
2. Let P be the shortest path from S to T. If the weight of every edge in the graph is incremented by 2, P will still be the shortest path from S to T.

F

1. 3

参考

<https://www.freecodecamp.org/chinese/news/dijkstras-shortest-path-algorithm-visual-introduction/>

1. The minimum spanning tree of any weighted graph \_\_\_\_D

A.must be unique

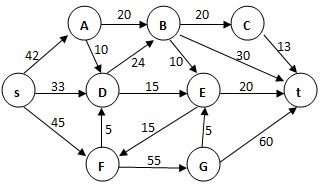
B.must not be unique

C.exists but may not be unique

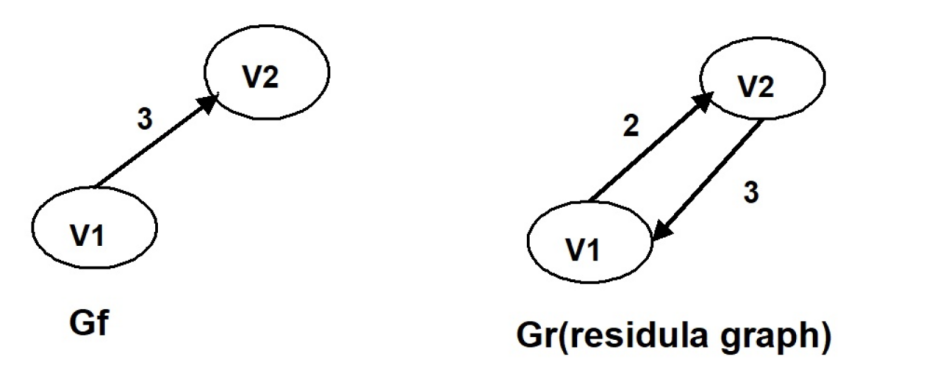
D.may not exist

因为没有说是联通的图。如果联通的话也不是唯一的，假如权值相同的矩形的最小生成树就有两种

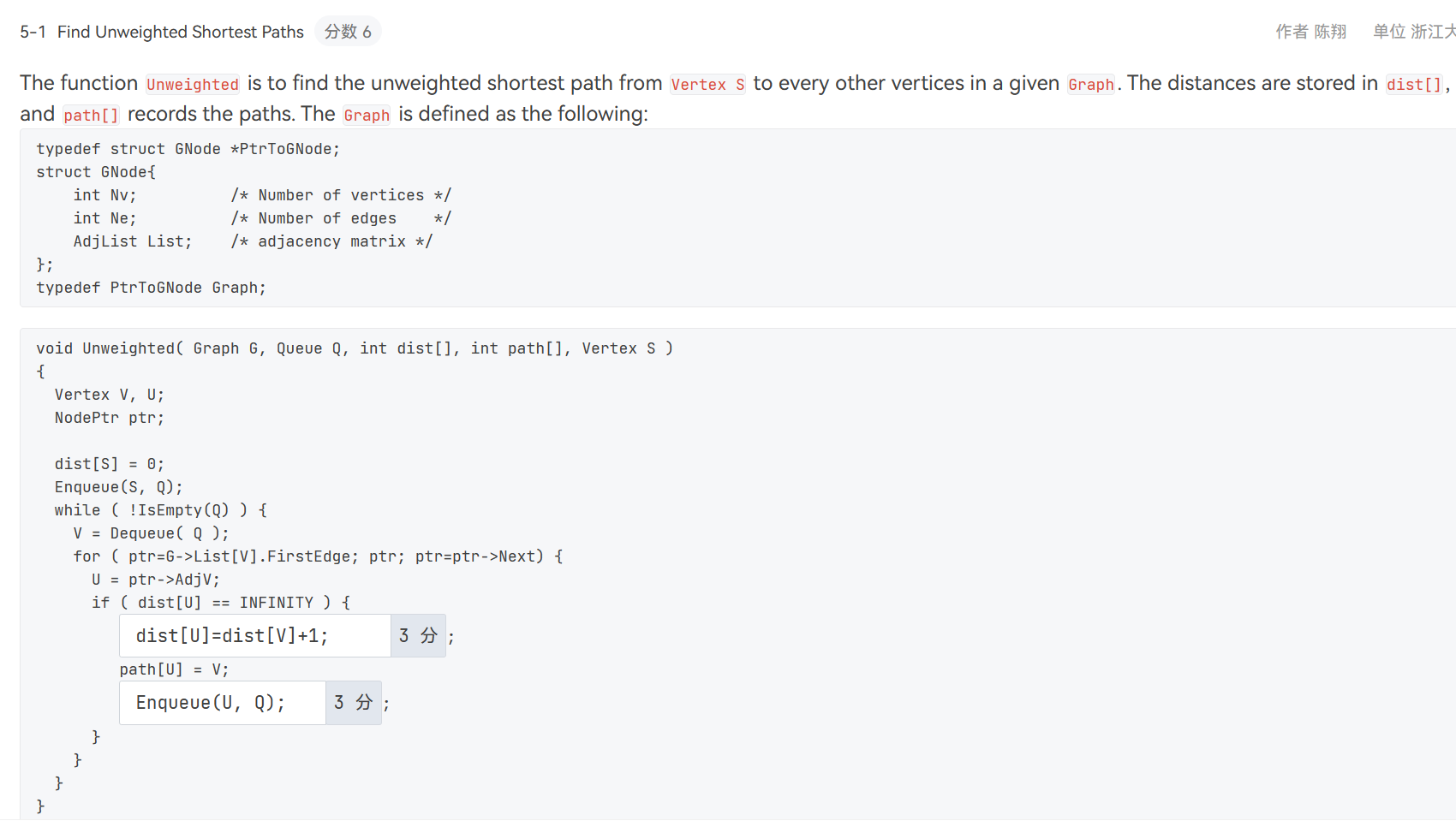
1. The maximum flow in the network of the given Figure is:

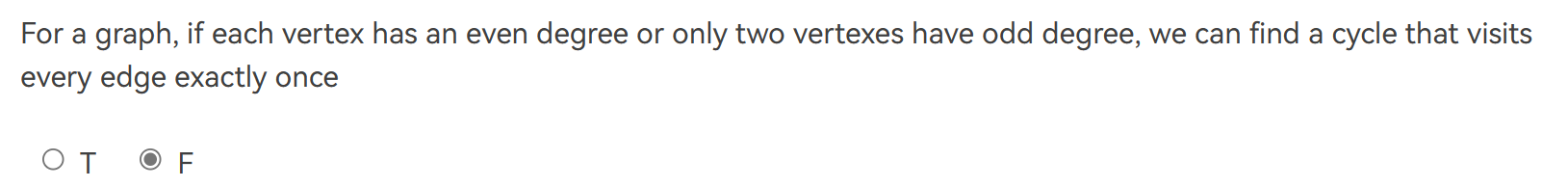


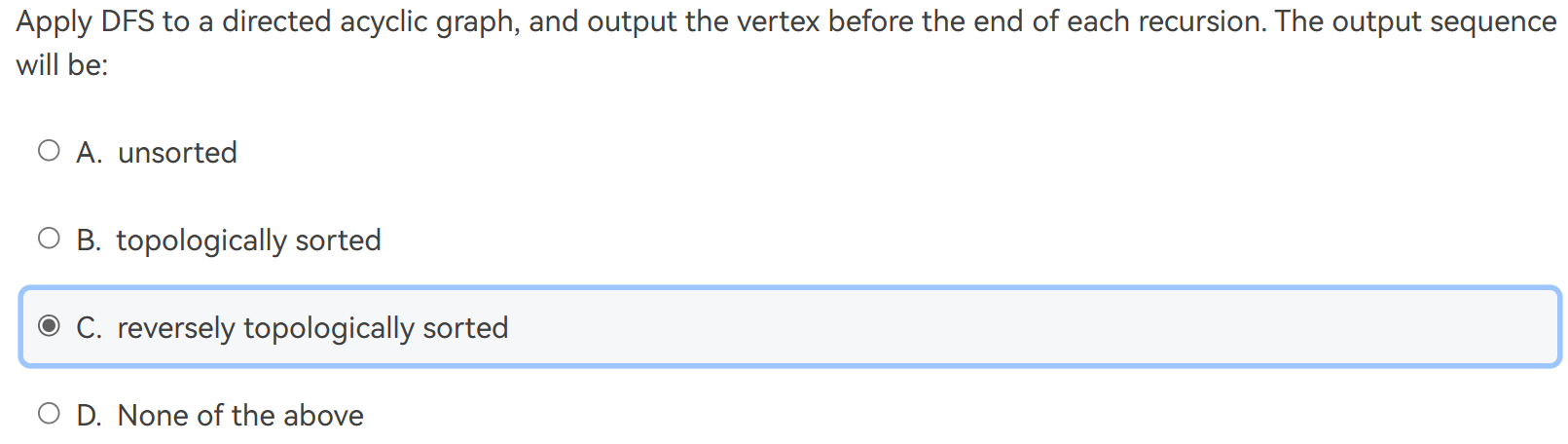
104 Gr法画图

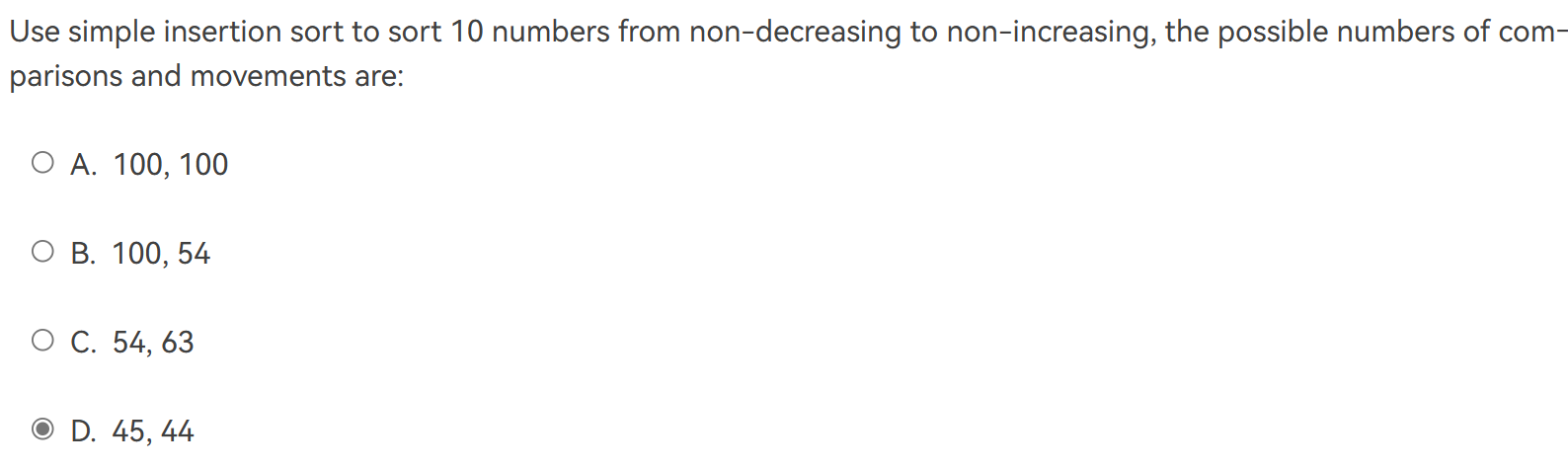
1. When solving the maximum flow problem for graph G, if partial states of the Gf​ ( will be the maximum flow when the algorithm terminates) and Gr​ (residual graph) are shown as the following, what must be the capacity of (v1, v2) or of (v2,v1) in the original graph G?

Gr中，每条边的双向流量和不变，答案是5.



1. 可能是非连通图

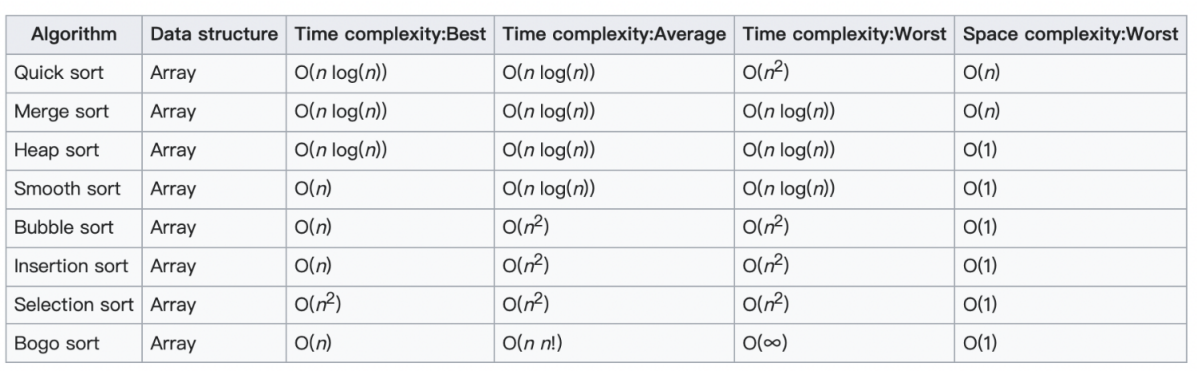
38.递归的顺序是遍历1->遍历2->遍历3->.....->遍历n->返回n->返回n-1...->返回3->返回2->返回1，返回的时候打印，因此是从最后一个结点开始打印到源点。是逆拓扑序

1. The number of comparisons won’t be larger than 45 逆序对就只有45个
2. If there are less than 20 inversions in an integer array, the Quick Sort will be the best method among Quick Sort, Heap Sort and Insertion Sort.

F 数组越大，快排的效果越好

1. During the sorting, processing every element which is not yet at its final position is called a "run". To sort a list of integers using quick sort, it may reduce the total number of recursions by processing the small partion first in each run.

F 递归深度只和pivot的选择有关，因为每次实际上只是找出了小于pivot的数字和大于pivot的数字，所以就算modify也不影响最后这些位置和pivot的相对大小，如果pivot每次都选最大或最小，递归深度是最大的。



1. During the sorting, processing every element which is not yet at its final position is called a "run". Which of the following cannot be the result after the second run of quicksort?
2. 5, 2, 16, 12, 28, 60, 32, 72
3. 2, 16, 5, 28, 12, 60, 32, 72
4. 2, 12, 16, 5, 28, 32, 72, 60
5. 5, 2, 12, 28, 16, 32, 72, 60

After the element is sorted, the right of the sorted element must be larger than the sorted element, the left must be smaller. 每一个run可以确定一个pivot的位置，两次递归可以确定1(run1)+2(run2因为左右各一次)=3个。但是要注意，如果在末尾的pivot使得递归只在之前进行(也就是只有左边的递归)，所以也可以。选D

1. Among the following sorting methods, which ones will be slowed down if we store the elements in a linked structure instead of a sequential structure?

1.Insertion sort; 2. Selection Sort; 3. Bubble sort; 4. Shell sort; 5. Heap sort

如果在链表中存储数据，变慢的操作是访问第n个元素，变快的是插入。Shell需要访问第k个元素，heap需要访问第i/2个元素，前三种顺序存储时也要遍历所以不影响

1. When selecting a sorting algorithm, which of the following factors must be taken into consideration besides the time and space complexities?

I、the size of input data

II、the structure used to store the data

III、the stability of the algorithm

IV、the initial condition of the data

都需要考虑

1. Which of the following statements about HASH is true?
2. the expected number of probes for insertions is greater than that for successful searches in linear probing method在线性探测方法中，插入的预期探针数大于成功搜索的探针数
3. if the table size is prime and the table is at least half empty, a new element can always be inserted with quadratic probing

如果表大小为质数，并且表至少为半空，则始终可以使用二次探测插入新元素

1. in separate chaining method, if duplicate elements are allowed in the list, insertions are generally quicker than deletions如果列表中允许重复的元素，则插入通常比删除更快

插入不需搜索整个链表，可直接添加到链表的末尾。删除需要搜索并定位要删除的元素

D.all of the above

1. 哈希表线性探测的搜索长度https://www.cnblogs.com/47Gamer/p/13160610.html
2. Given an initially empty hash table HT of size 11. The hash function is H(key)=key% 7, with linear probing used to resolve the collisions. Now hash the keys 87, 40, 30, 6, 11, 22, 98 and 20 one by one into HT. What is the average search time for unsuccessful searches?

6

查找不成功的搜索次数是不成功次数(题目中也就是地址0~6对应的不成功次数之和具体可以参考上面网址)/mod后面跟的数字(题目中是7)

1. Given an initially empty hash table with length 7, together with a hash function HTH(k)=k%7. Let us use linear probing to solve collisions. What is the average search length for successful searches after inserting 22, 43, 15 one by one into ?

2

查找成功的情况是搜索成功次数(题中是1+2+3)和/查找成功的次数(题目中就是3个数)

1. Given input {4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function h(X)=X%10. If the collisions are solved by open addressing hash table with second hash function h2​(X)=7−(X%7) and table size being 10, then the indices of the input numbers in the hash table are: (-1 means the insertion cannot be successful)

1 3 4 9 7 5 -1

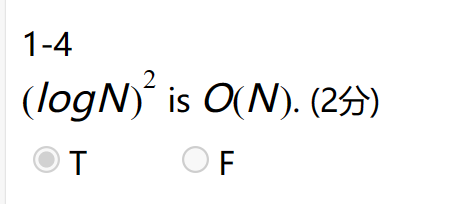
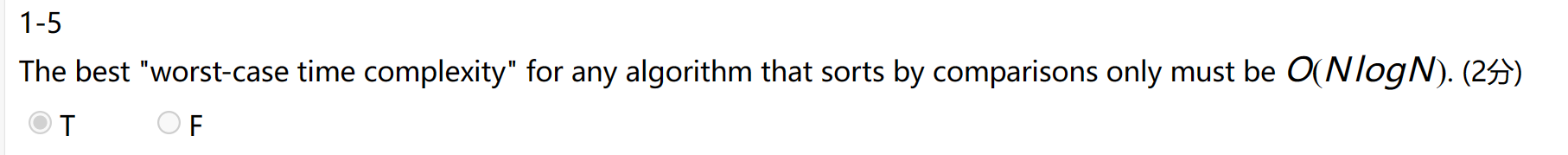
6173的原始哈希值是3冲突，h2(6173)=1，然后f ( i ) = i \* hash2( x )是1，而h(6173)=3，所以( h(x) + f(i) ) % TableSize计算位置为4；同样的4344得出原始哈希值是4冲突，h2(4344)=3，然后f ( i ) = i \* hash2( x )是3，用( h(x) + f(i) ) % TableSize计算位置为7；最后一个1989由于h2(1989)=6，f ( i ) = i \* hash2( x )是6，( h(x) + f(i) ) % TableSize计算位置为5又冲突了，继续i++，f ( i ) = i \* hash2( x )是12，( h(x) + f(i) ) % TableSize计算位置为1又冲突了，继续i++，f ( i ) = i \* hash2( x )是18，( h(x) + f(i) ) % TableSize计算位置为7又冲突了，继续i++，f ( i ) = i \* hash2( x )是24，( h(x) + f(i) ) % TableSize计算位置为3又冲突了，继续i++，f ( i ) = i \* hash2( x )是30，( h(x) + f(i) ) % TableSize计算位置为9得到循环所以不行

1. Suppose that the numbers {4371, 1323, 6173, 4199, 4344, 9679, 1989} are hashed into a table of size 10 with the hash function h(X)=X%10, and hence have indices {1, 3, 4, 9, 5, 0, 2}. What are their indices after rehashing using h(X)=X%TableSize with linear probing?

1, 12, 9, 13, 20, 19, 11

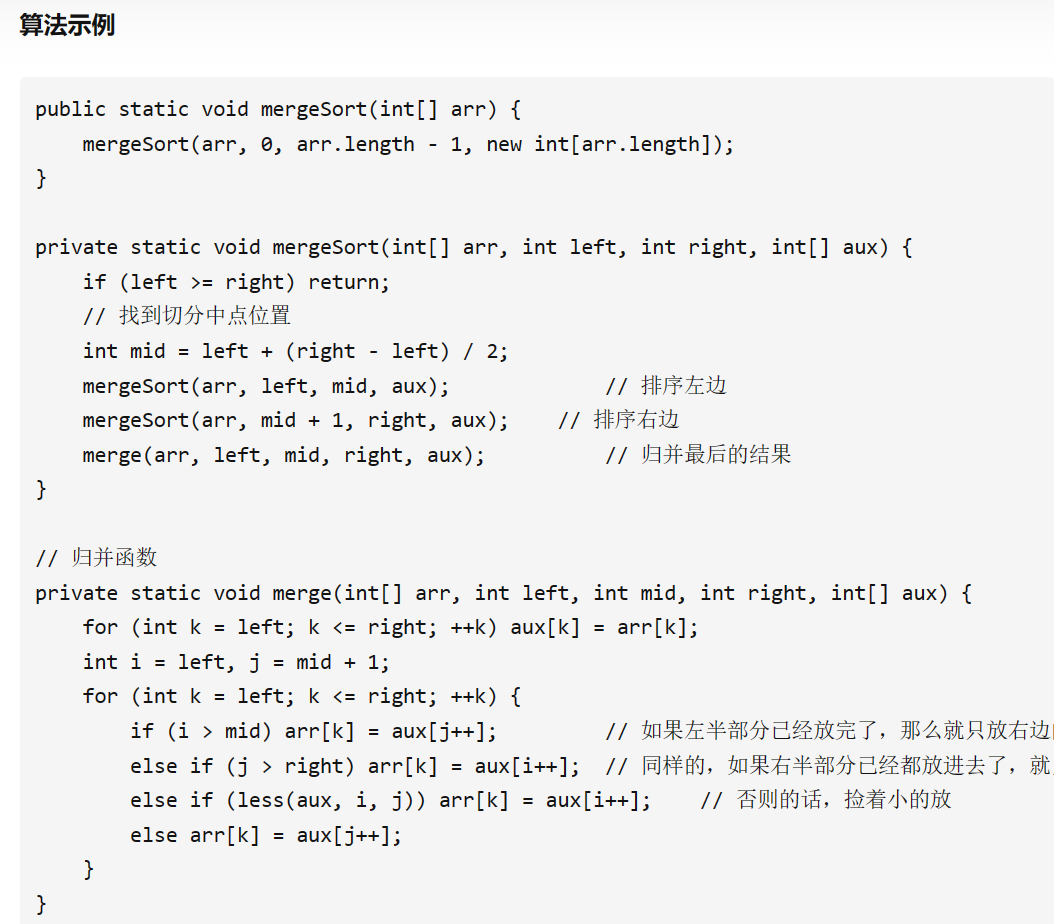
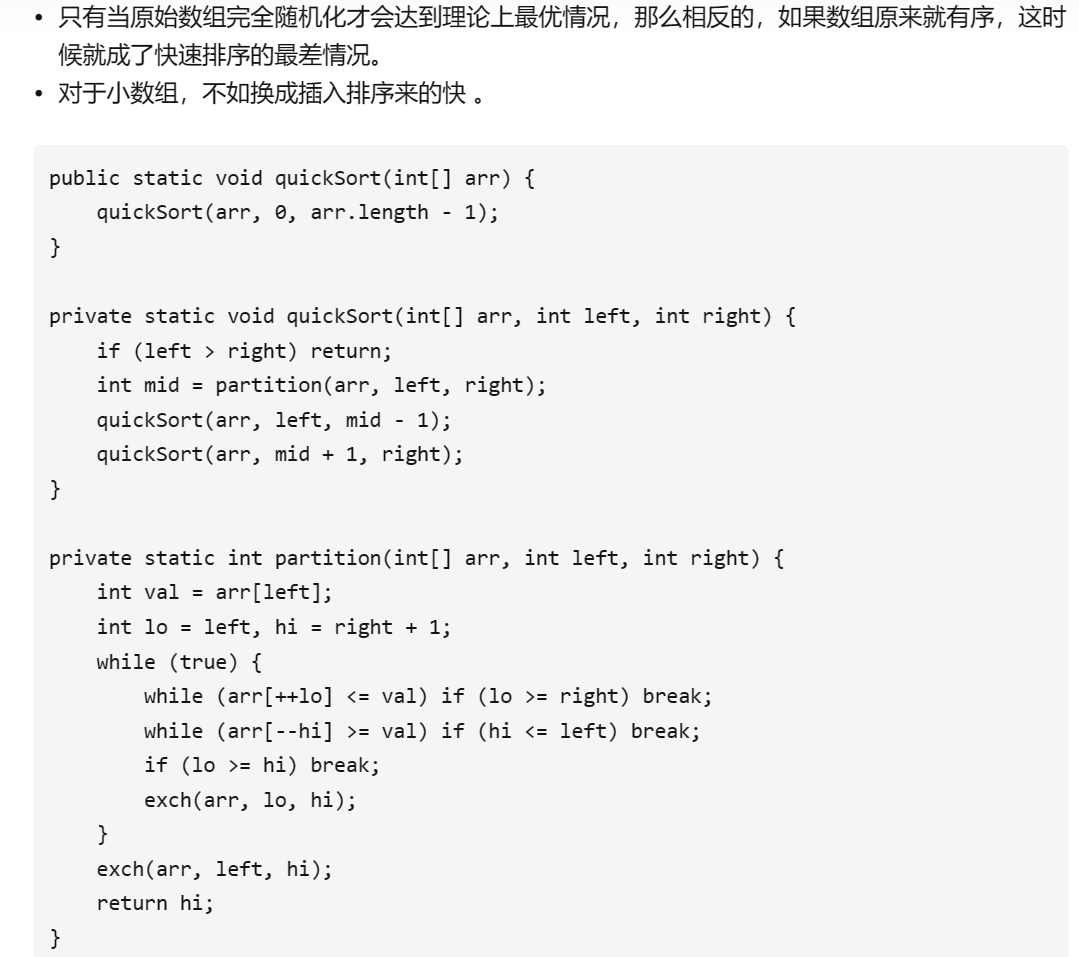
再散列将散列表扩大一倍，表长变为20，然后取最近的质数23作为表长

期中考试整理：



冒泡排序、选择排序、插入排序：O(N^2)

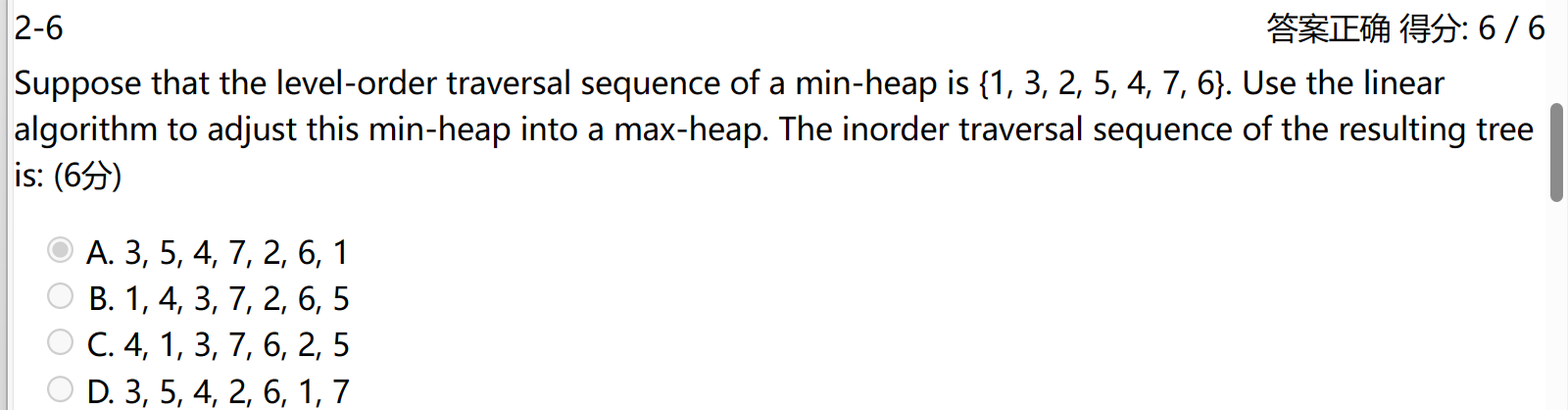
归并排序：时间复杂度O(NlogN) 空间复杂度O(N)

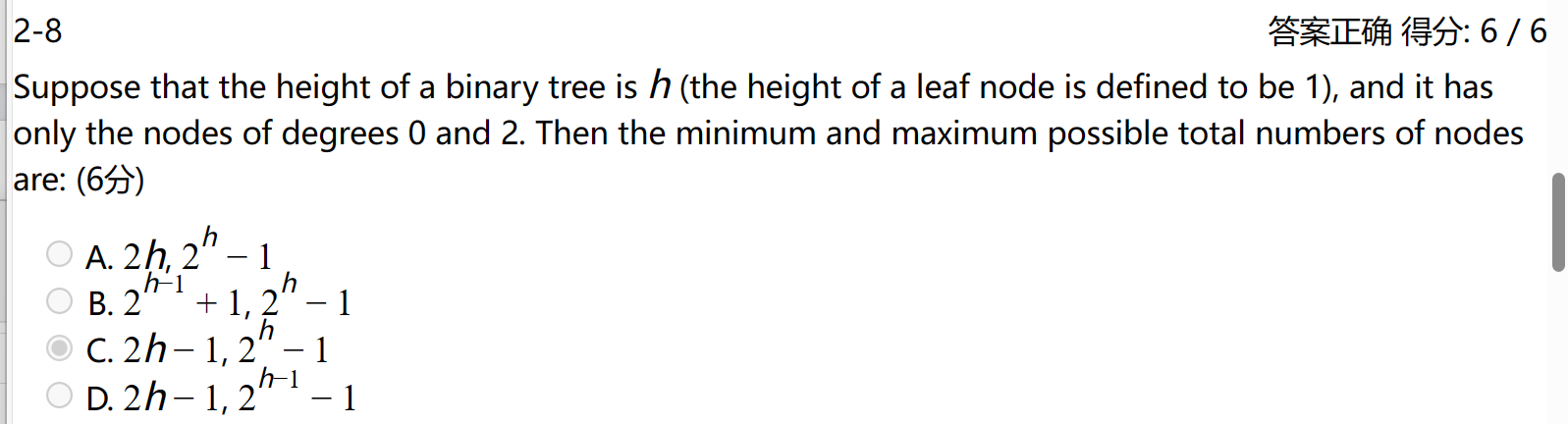
快速排序：最坏时间复杂度O(N^2) 平均时间复杂度O(NlogN)



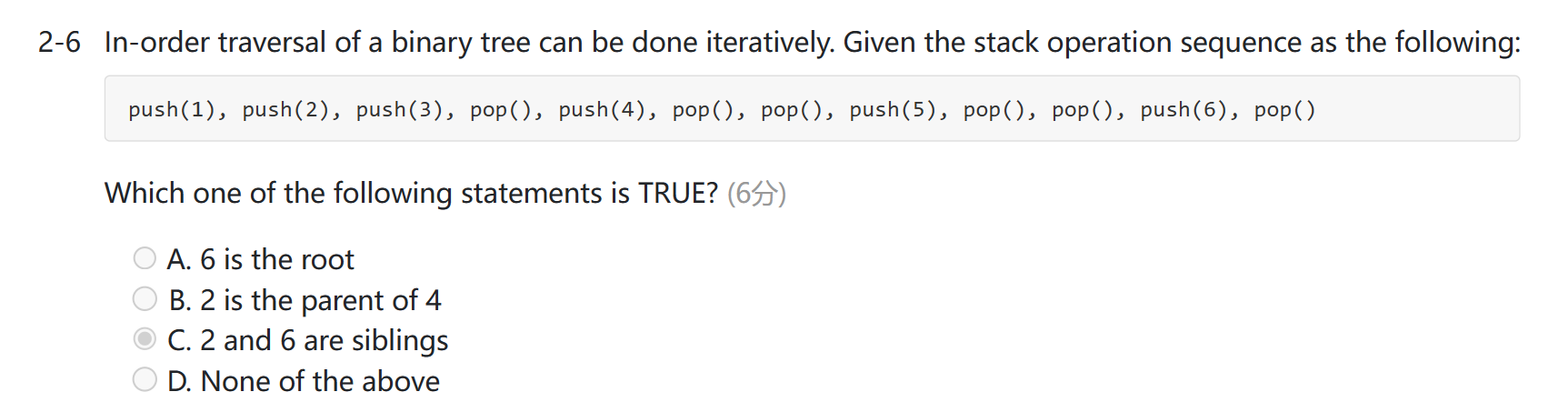
O(N2)

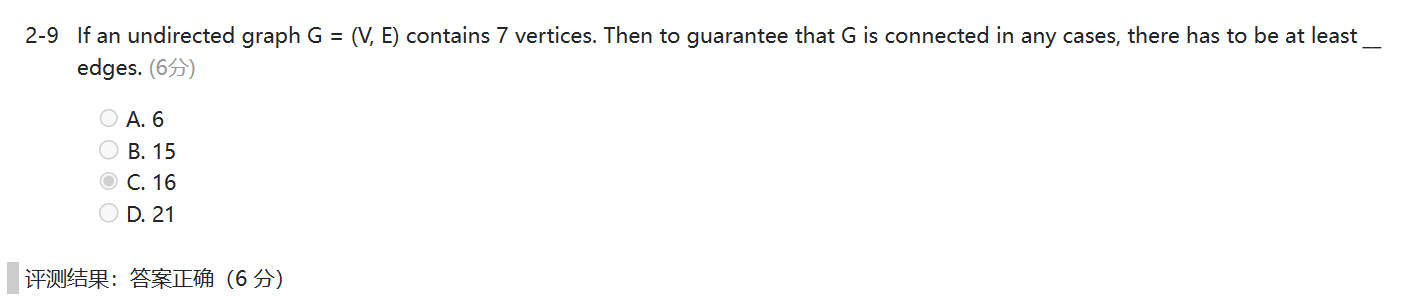
比较排序

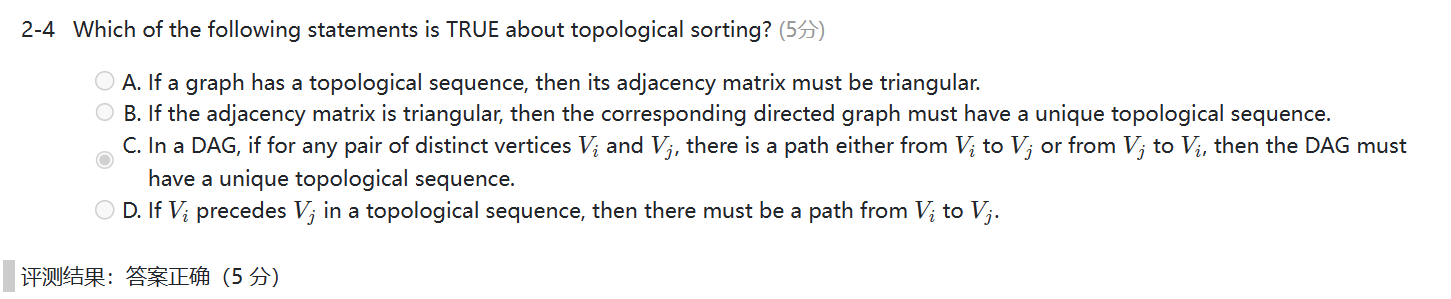
Linear算法就是从叶子节点开始重新排序，然后一层一层向上，直到根结点变成max后再一层一层向下调整大小。

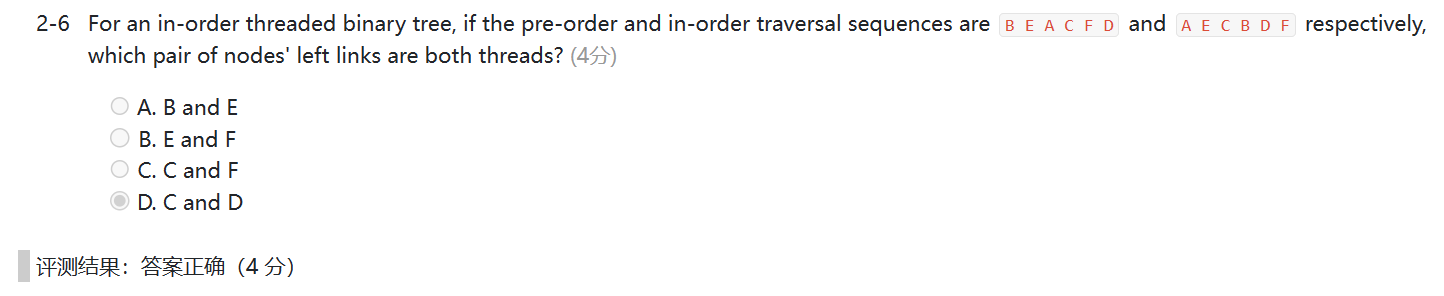
（A在以下排序方法，哪一种可能导致在最后一次运行开始之前没有一个元素处于其最终位置？假设有 2 个以上的元素需要排序）

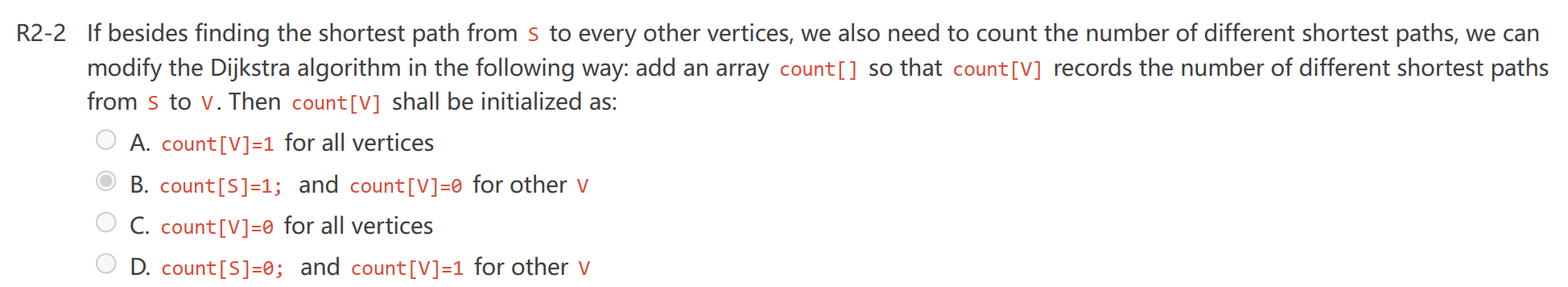
15-16qz函数题6-1

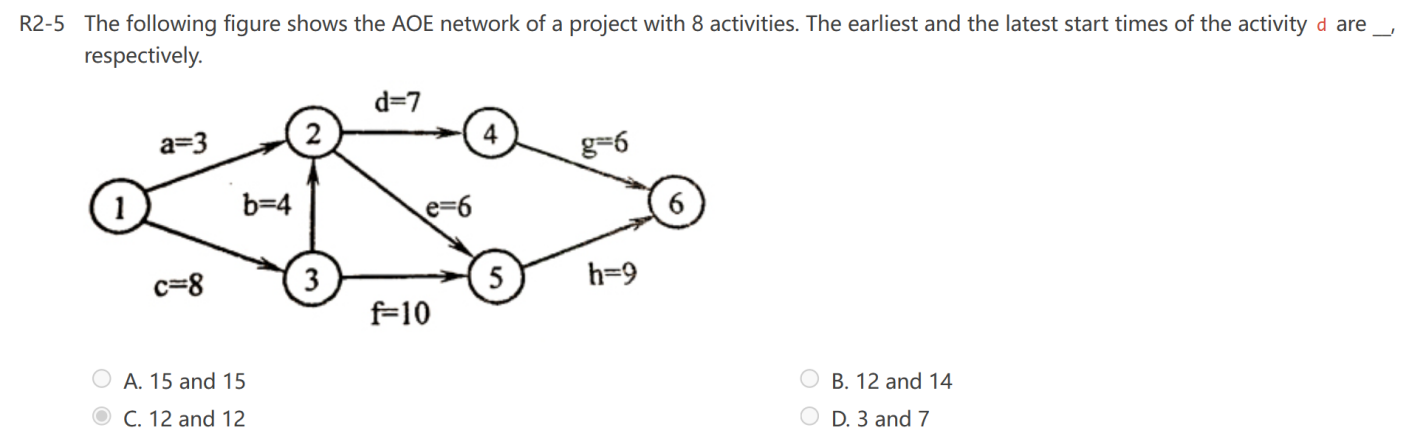


其他6 个点完全连接，再加一条边

有向图的邻接矩阵不一定是三角矩阵和对称矩阵，比如一个拓扑排序为231，那么它的矩阵就是非三角和对称的。三角矩阵说明该图是一个无环图，不能说明它的拓扑序列是唯一的。D选项VI先于vj，但是他俩可能没有关联。DAG图是无环图，任何两个点都有一条路径，说明每个点的出度不会是二，即唯一的拓扑排序

Left links是threads的意思是，没有左孩子，但是将左节点连到中序排序的前一个节点的位置，这也是线索树的定义，这题画出二叉树之后，c和d都没有左孩子，所以他们左边的结点指向的是thread

初始化count[S]为1是为了确保从源点S到自身的最短路径数量为1，而其他顶点的count值初始化为0表示尚未找到任何最短路径

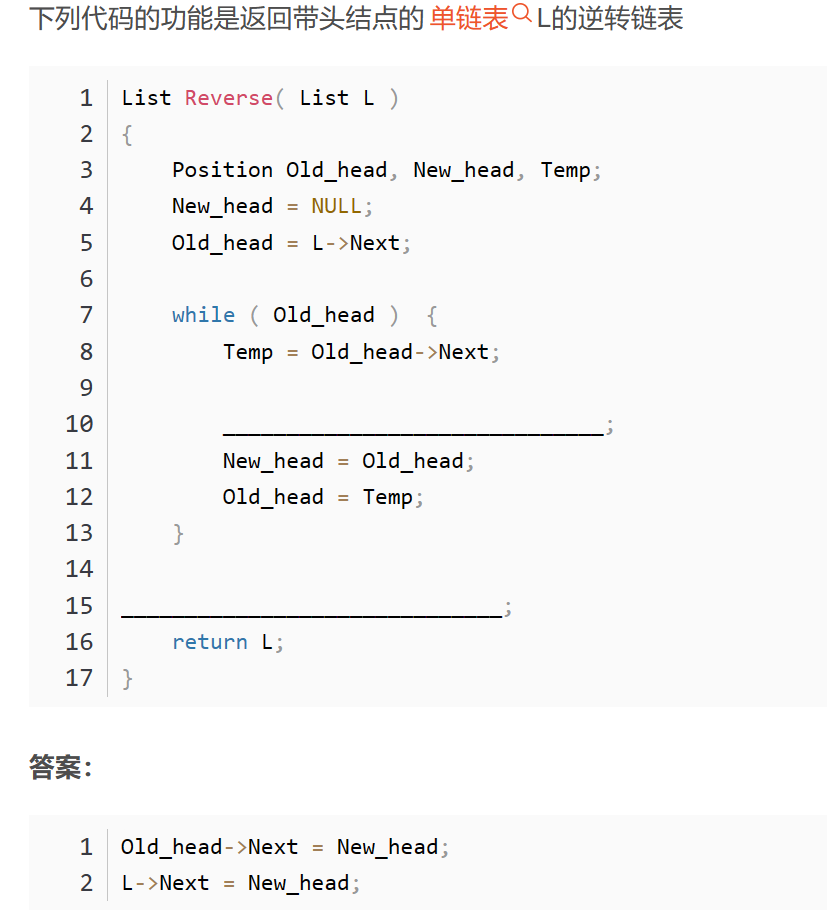


选B？？

参考解析https://zhuanlan.zhihu.com/p/337438327

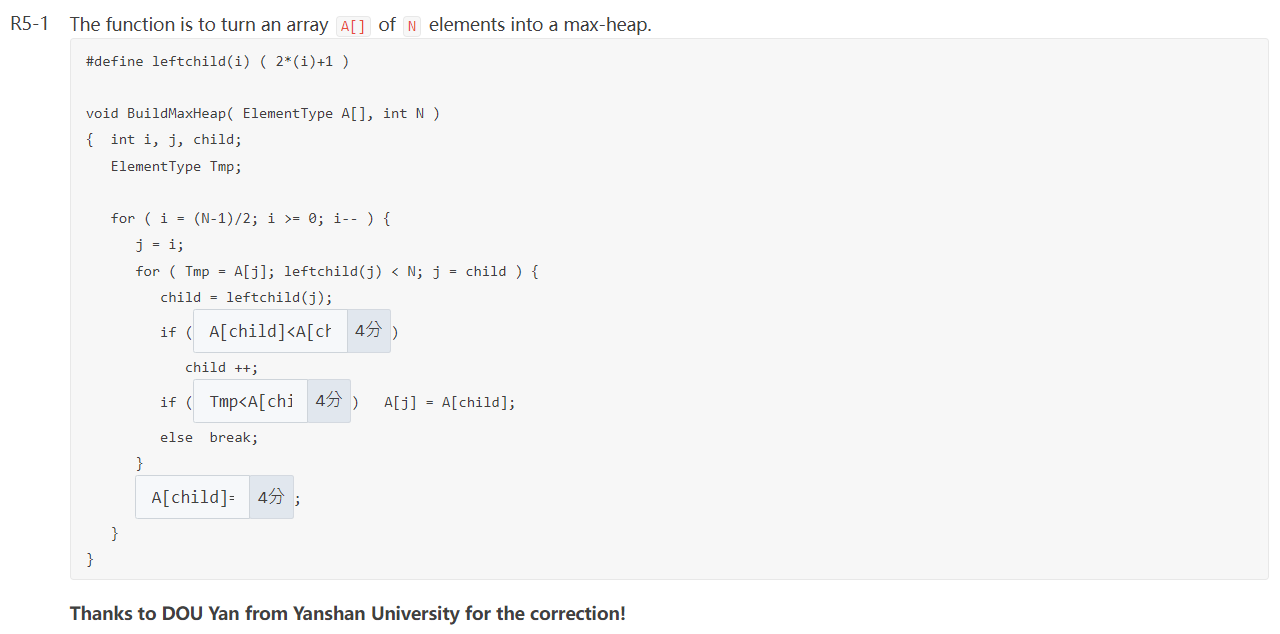
最早发生时间：从前往后，前驱结点到当前结点所需时间，取最大值

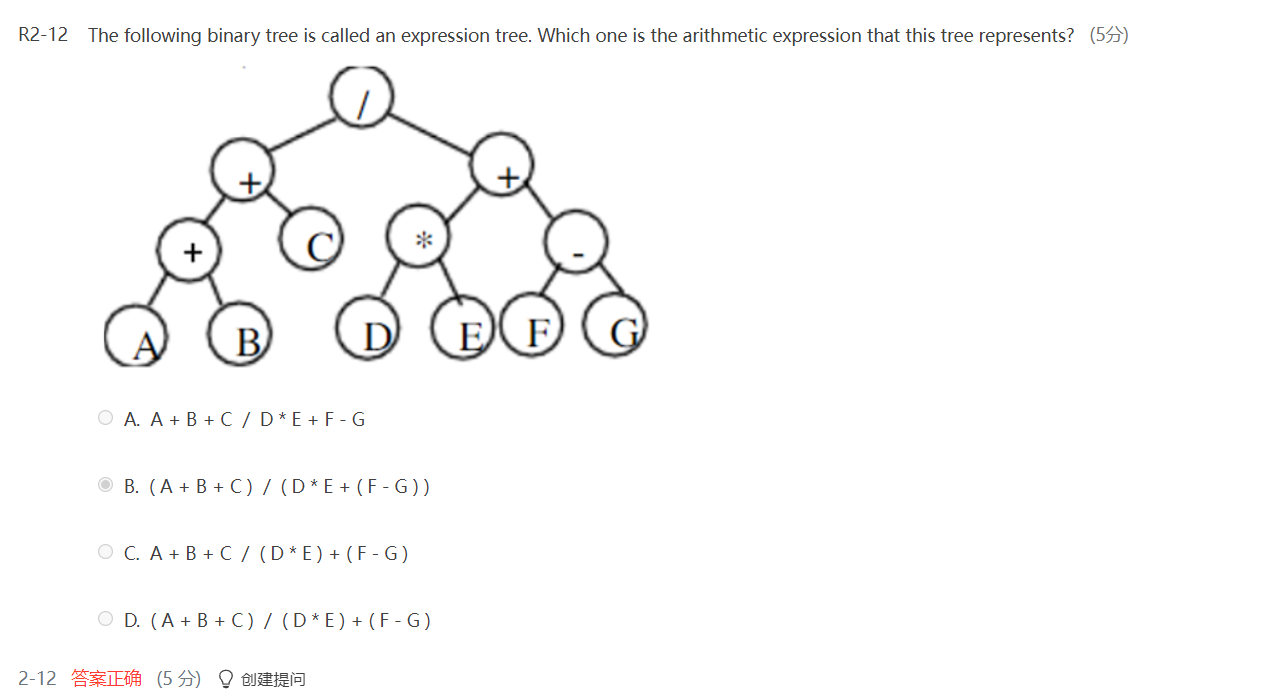
最迟发生时间：从后往前，后继结点的最迟发生时间-边权值，取最小值。

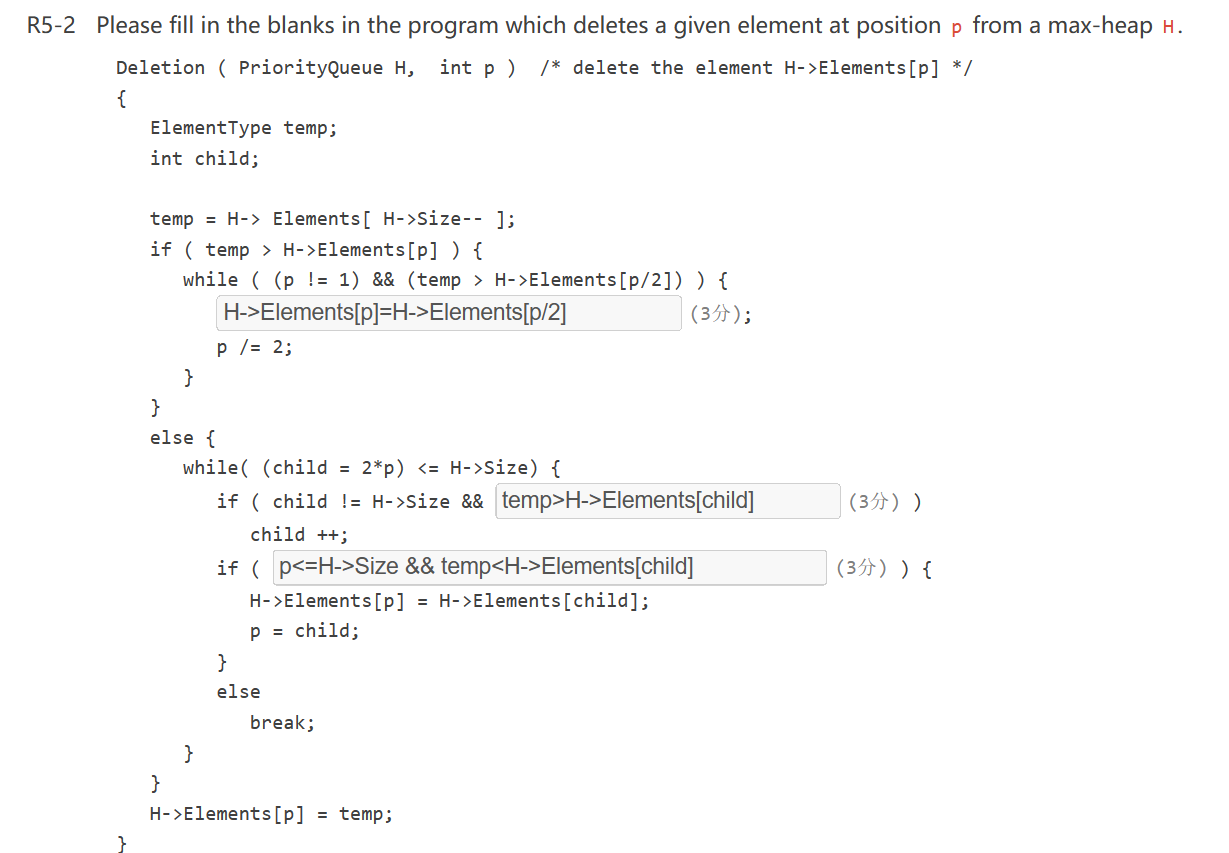


5-1 第1空：child!=N-1 && A[child+1]>A[child]

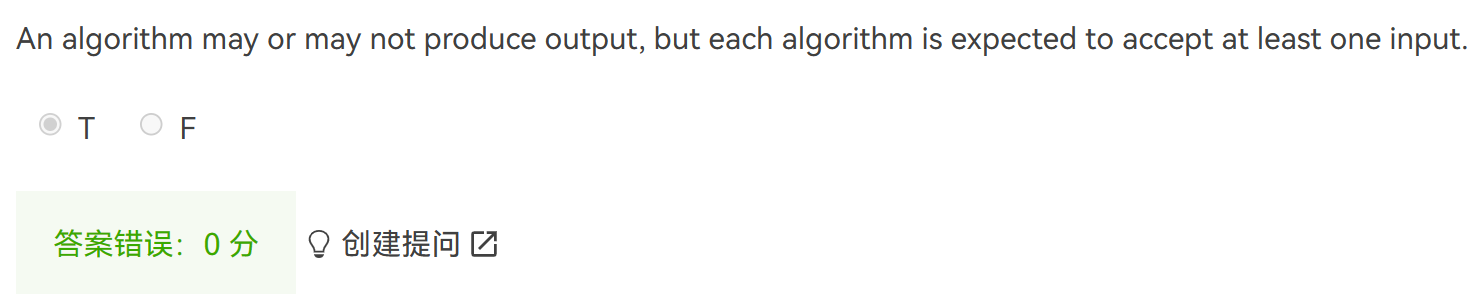
第2空：Tmp < A[child]

 第3空：A[j] = Tmp



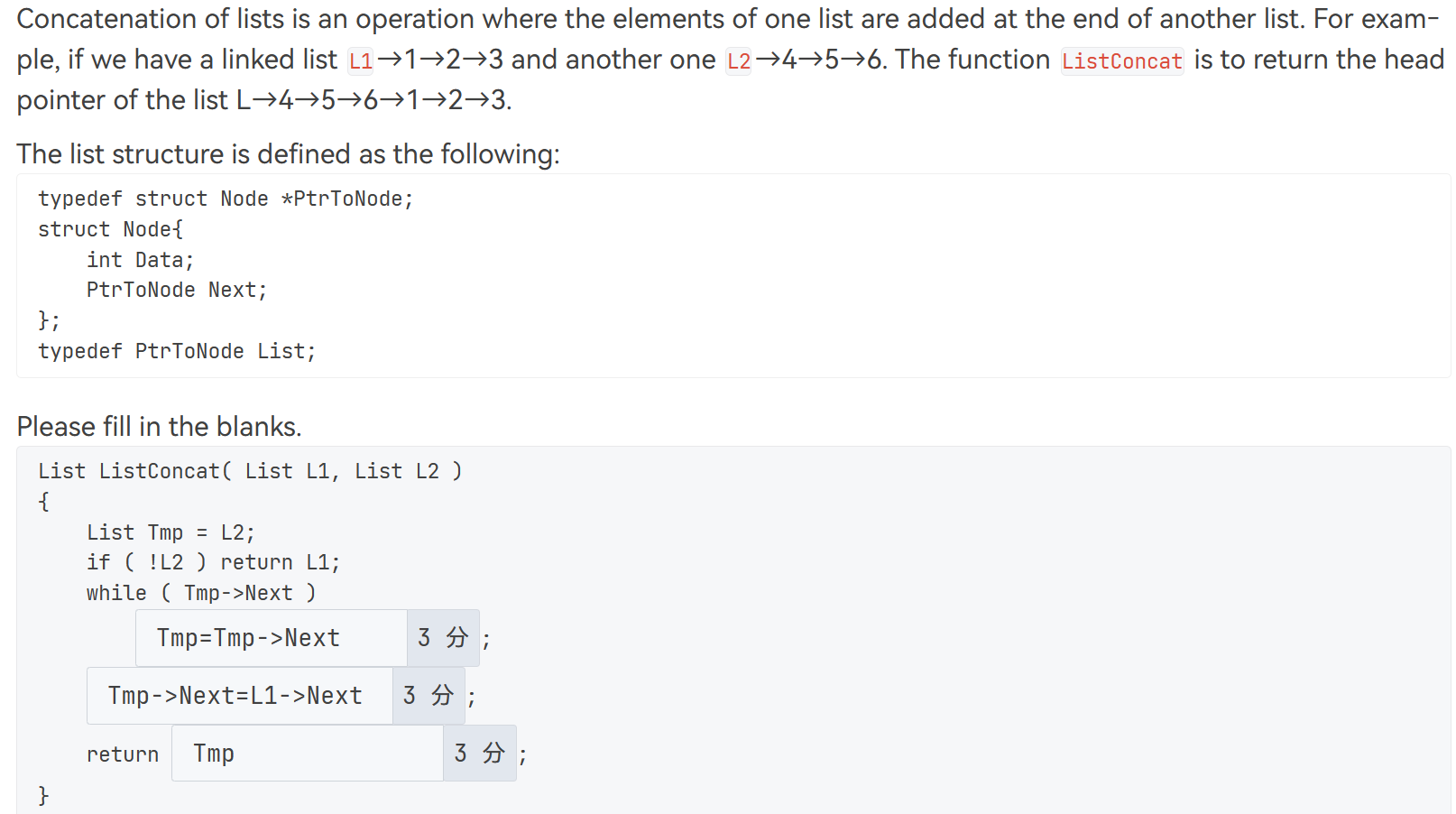
1. >Elements[p]=H->Elements[p/2]

child != H->Size && (H->Elements[child+1] >H->Elements[child])

temp < H->Elements[child]

输入input可以是0个元素

Set name就不是root根节点了



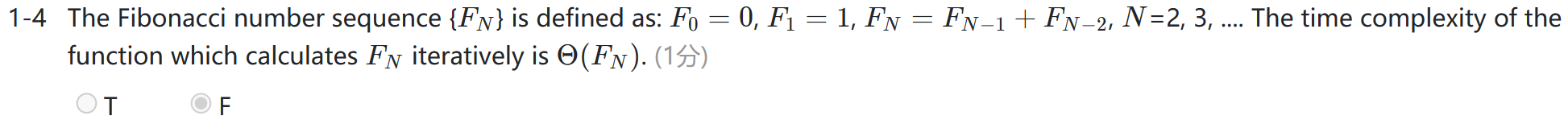
Tmp->Next = L1

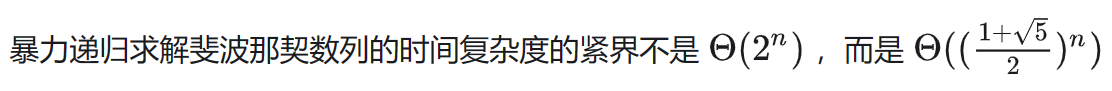
L2

期末：

1. If a graph is represented by adjacency lists, then the space taken depends only on the number of vertices, not the number of edges.

F





->FN



斐波那契数列 用递归法时间复杂度O(n2) 也就是题目的；空间复杂度O(1)

用迭代的方法也就是循环法时间O(n) 空间O(n)

1. To sort N records by merge sort, the number of merge runs is O(NlogN)

或者To sort N records iteratively by merge sort, the number of runs is O(NlogN)

都是F merge runs O(logN) 注意the number of runs和时间复杂度是不一样的

1. In hashing, functions "insert" and "find" have the same time complexity.

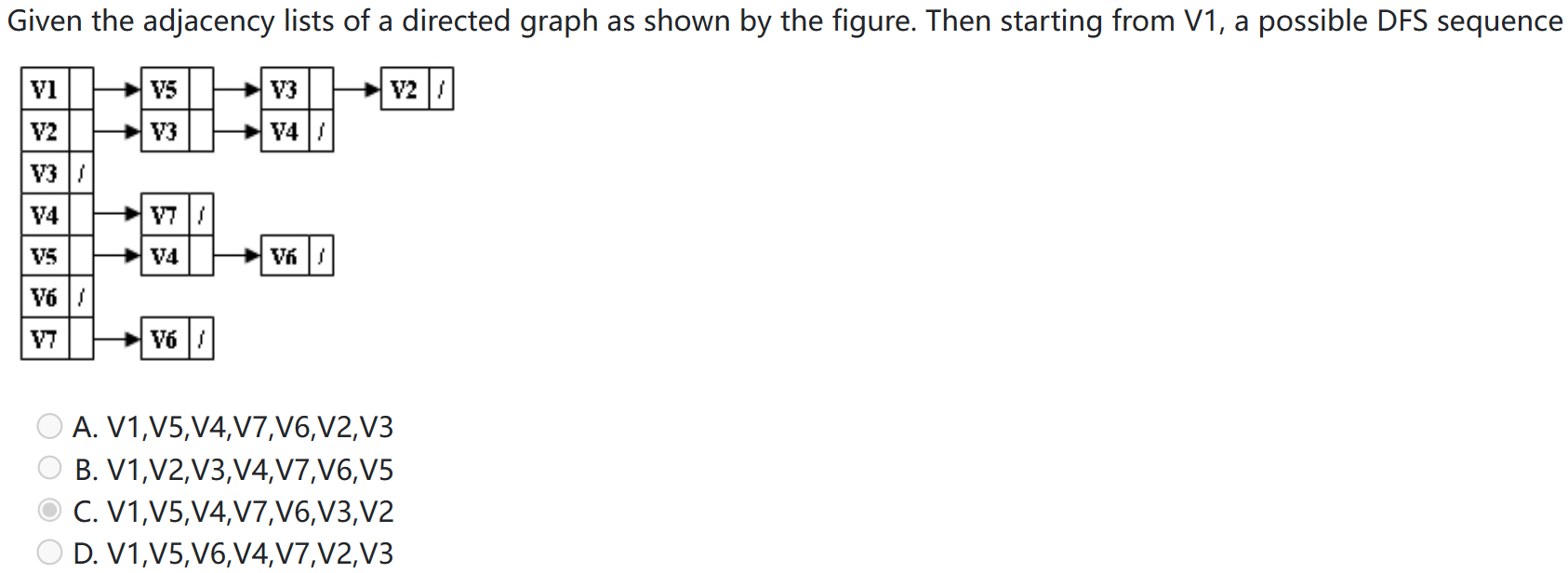
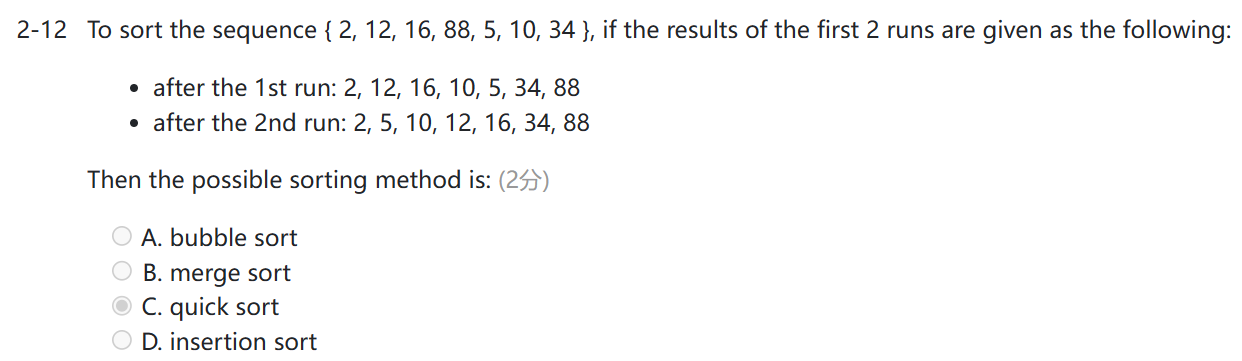
T 都是O(1)

1. In a graph G, if we have to do BFS twice to visit every one of its vertices, then there must be a cycle in G.

F 如果是连通图则只需要一次BFS就能遍历所有点

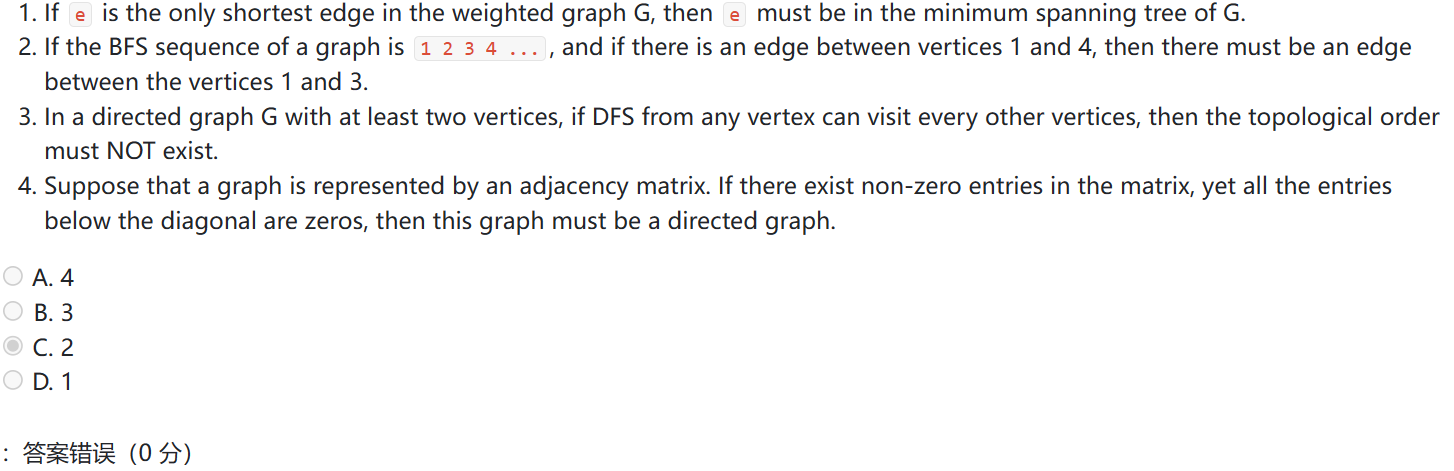
1. Given a hash table of size 17 with the hash function H(Key)=Key%17. Quadratic probing (hi(k)=(H(k)±i2)%17) is used to resolve collisions. Then after inserting { 23, 22, 7, 26, 9, 6 } one by one into the hash table, the address of 6 is

2 这里要注意是正负号也就是说可以是6-22=2

1. 要根据邻接表的顺序（例如回溯的时候先V3后V2）
2. 
3. Given the result of the 1st run of a sorting method as { 19, 21, 7, 14, 5, 27, 1, 10 }. Then among the following, this method has to be:

Merge sort

1. DFS用于检测一个有向图中是否存在环
2. 其中1、3、4是正确的，2可以由下图表示;3如果从有向图 G 的每一点均能通过深度优先搜索遍历到所有其它顶点，则该图是一个有环图



1

/ \

4 2

/

3

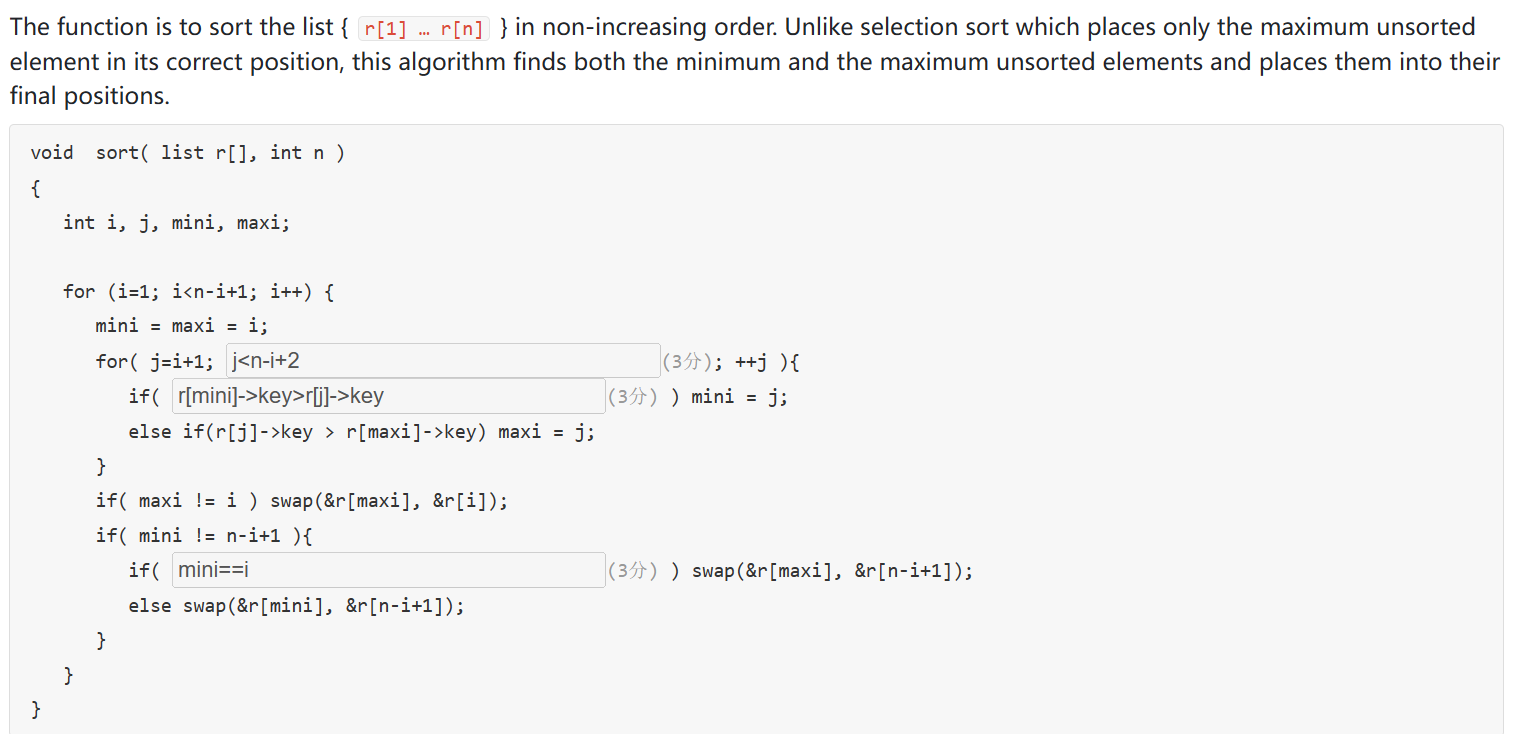
1. If quadratic probing is used to resolve collisions, then which one of the following statements is TRUE about inserting a new element? (2分) (B)

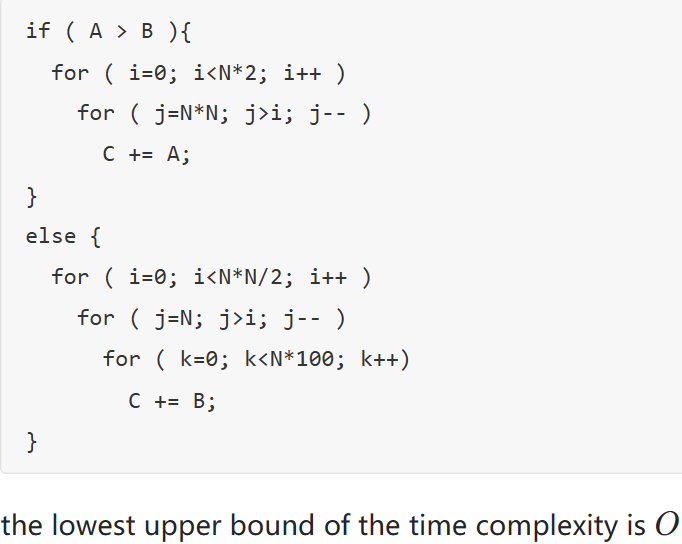
A.The insertion will be successful. B.The insertion may not be successful.

C.The insertion will not be successful.

D.The insertion will be successful if the size of the table is prime.

插入操作的成功与哈希表中是否还有空槽可供使用有关，而与哈希表的大小是否为质数无关

13.

1. 函数题6-1：15-16qm CheckBST[2]
2. 如果条件 A > B 为真，外部循环的迭代次数为 N \* 2，内部循环的迭代次数为 N^2 - i。因此，内部循环的总迭代次数为：N^2 + (N^2 - 1) + (N^2 - 2) + ... + 1 = N^2 \* (N^2 + 1) / 2，时间复杂度为 O(N^4)；如果条件 A > B 为假，则有三个嵌套的循环。外部循环的迭代次数为 N^2 / 2，第二个循环的迭代次数为 N - i，第三个循环的迭代次数为 N \* 100。因此，第三个循环的总迭代次数为：N \* 100 \* (N - 0) + N \* 100 \* (N - 1) + N \* 100 \* (N - 2) + ... + N \* 100 \* 1 = N \* 100 \* (N + 1) \* N / 2，时间复杂度为 O(N^3)
3. Prim's algorithm is to grow the minimum spanning tree by adding one edge, and thus an associated vertex, to the tree in each stage.

T

1. An algorithm to check for balancing symbols in an expression uses a queue to store the partial expression

F 用stack来储存，balancing symbols指的是括号等左右是否匹配

1. If X and Y are both leaf nodes in a binary tree, then there exists a binary tree with preorder traversal sequence ...X...Y... and postorder traversal sequence ...Y...X....

F 因为不管是pre还是post都是从左到右，不会出现X Y顺序反向出现的情况

1. Which of the following statements is TRUE about topological sorting? (3分) C

A.If a graph has a topological sequence, then its adjacency matrix must be triangular.

B.If the adjacency matrix is triangular, then the corresponding directed graph must have a unique topological sequence.

C.In a DAG, if for any pair of distinct vertices Vi and Vj, there is a path either from Vi to Vj or from Vj to Vi, then the DAG must have a unique topological sequence.

D.If Vi precedes Vj in a topological sequence, then there must be a path from Vi to Vj.

注意有向图中环的定义！DAG图是有向无环图

1. Which of the following statements is FALSE about the shortest path algorithms? (3分) C

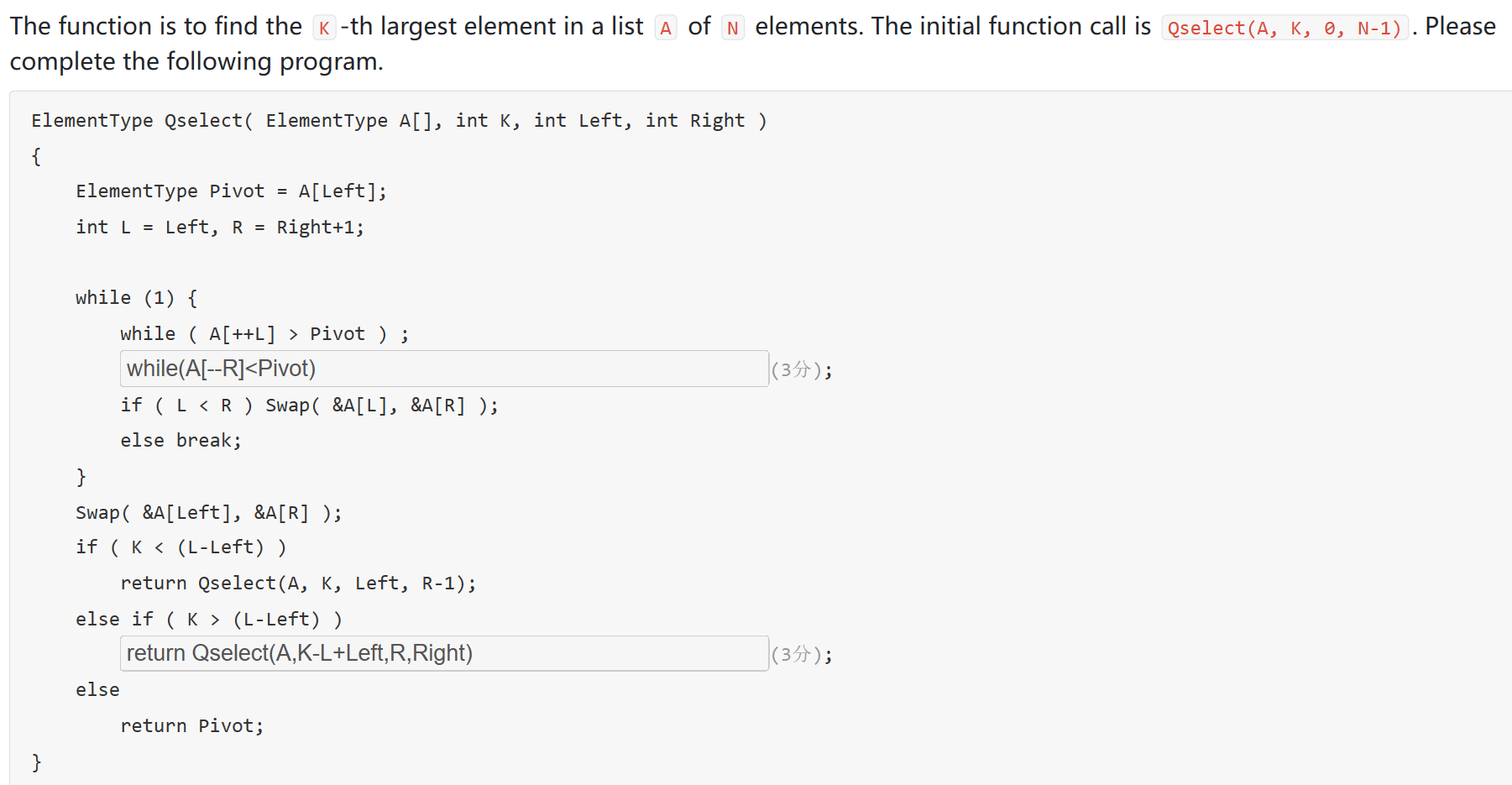
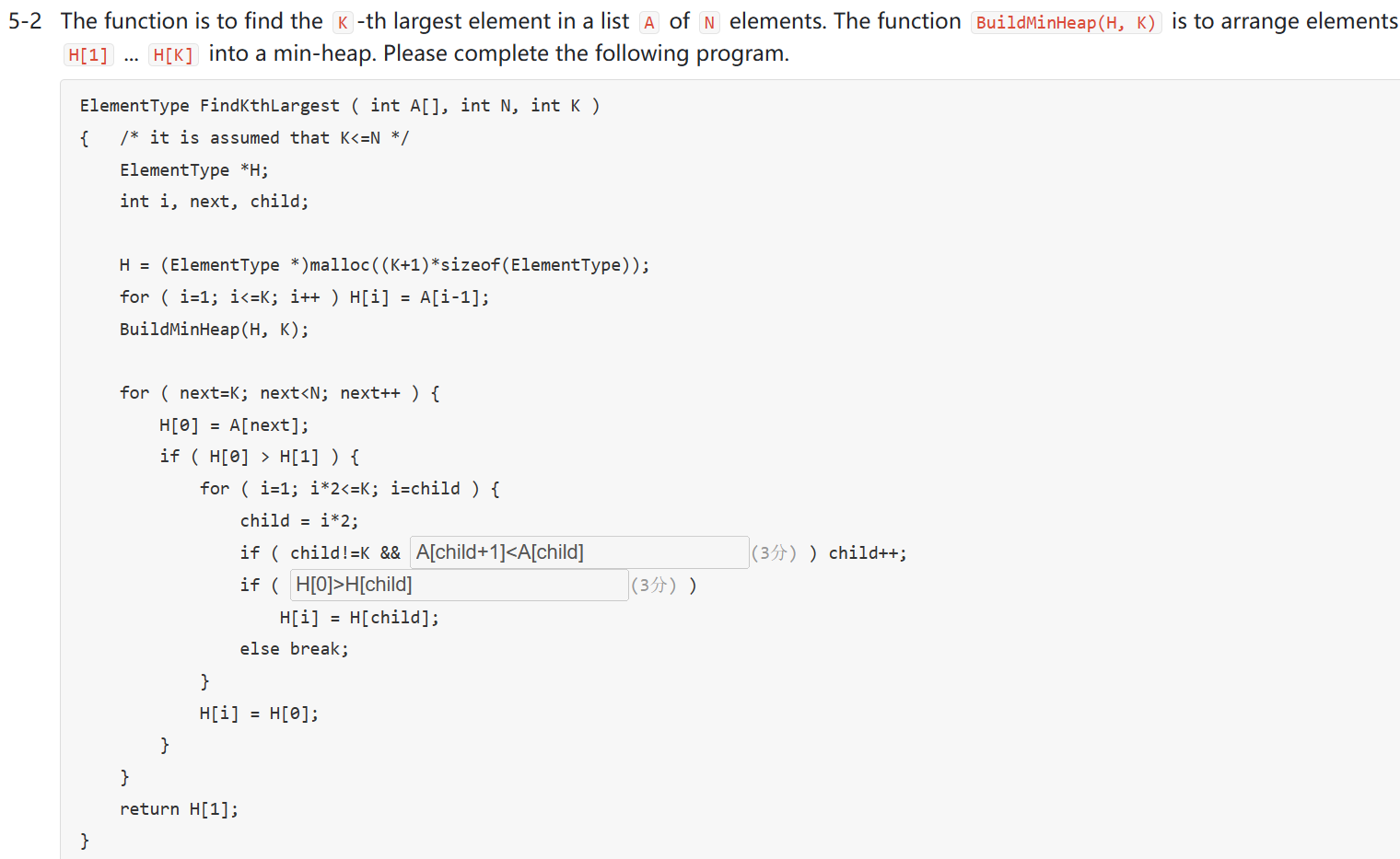
A. Dijkstra's algorithm works out the shortest path in increasing order of distances.

1. Single-source algorithms work fast on sparse graphs when finding the shortest path between any pair of vertices.单源算法（如Dijkstra算法和Bellman-Ford算法）的时间复杂度取决于图中的边数和顶点数。对于稀疏图来说，边数较少，单源算法会更快。
2. Negative-cost edge in a directed graph will cause infinite loop in Dijkstra's algorithm.

负权重边是指在图中存在一条边，其权重为负数。负循环是指在图中存在一个环（循环），其权重之和为负数。这意味着通过循环的重复遍历可以无限次地减小路径的总权重。负权重边可以导致错误的结果；而负循环更为严重，会导致算法陷入无限循环，无法终止

D. BFS can work out the shortest path when costs of all the edges are equal.

由于边的权重相等，BFS在每一层的节点中选择下一个节点时，会按照节点的先后顺序进行遍历。这就确保了当BFS到达目标节点时是通过最短路径到达，按层次逐步扩展的。

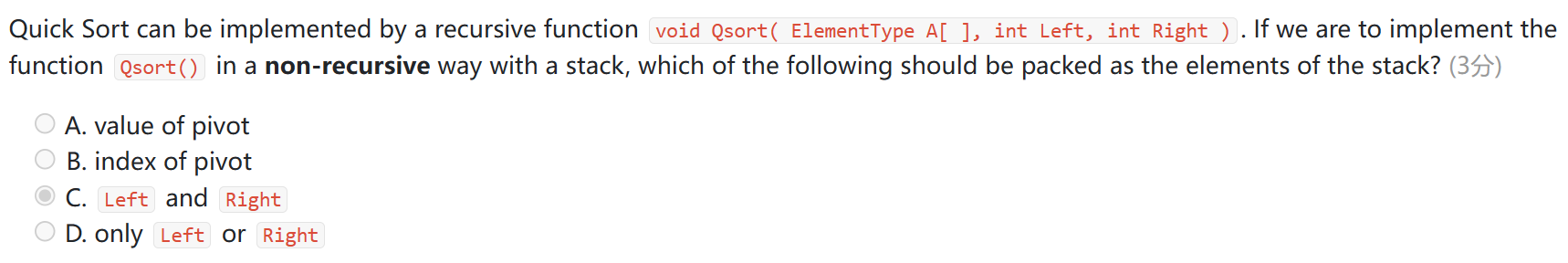
1. 这里第二个空格是return Qselect(A,K-L+Left,R+1,Right)
2. 第一个空是把A换成H...
3. 函数题6-1：16-17qm LCA in BST
4. The minimum spanning tree of a connected weighted graph with vertex set V={ A, B, C, D, E } and weight set W={ 1, 3, 2, 5, 1, 7, 9, 8, 4} must be unique.

T

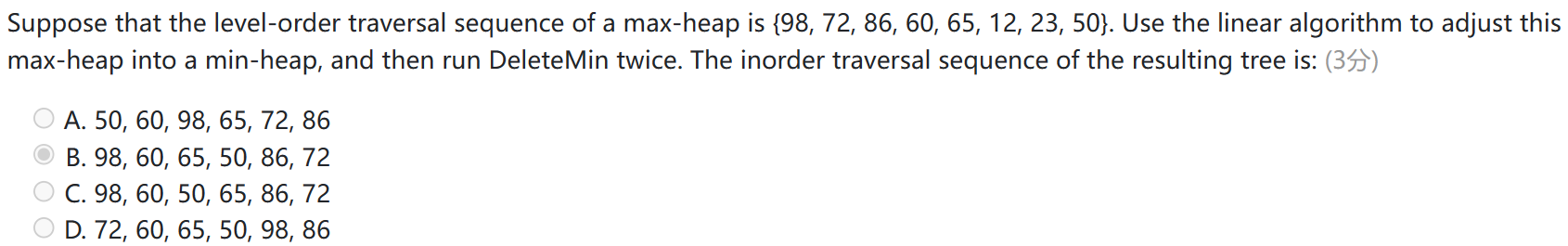
1. If 7 elements have been stored in a hash table of size 13 at positions { 0, 1, 2, 4, 5, 10, 11 }, and the hash function is H(x)=x%13. Then an empty spot can't be found when inserting the element 40 with quadratic probing.

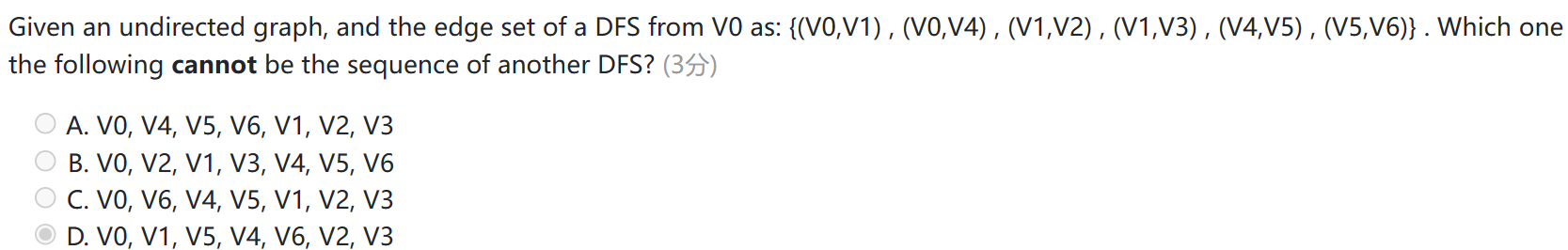
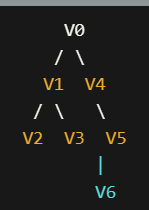
T

1. 
2. 在使用堆栈实现快速排序时，我们需要跟踪需要进行排序的子数组范围。每个范围可以由其左右索引表示。



1. 注意先从下往上移动然后再从上往下调整



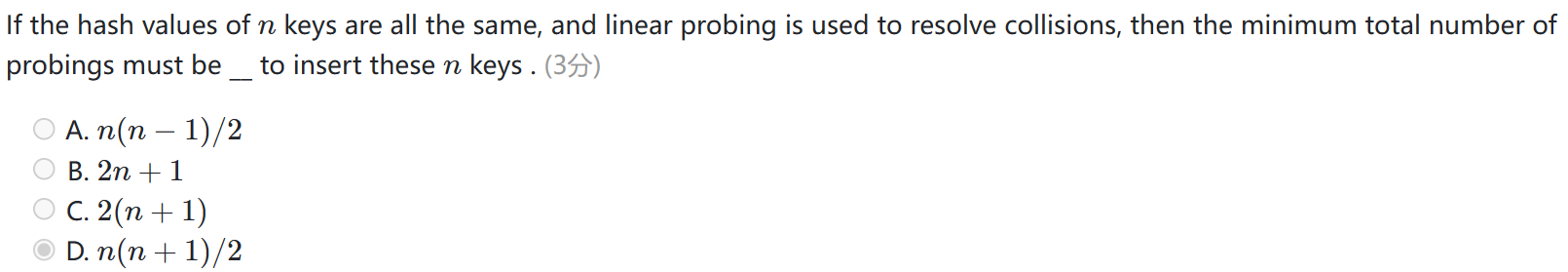
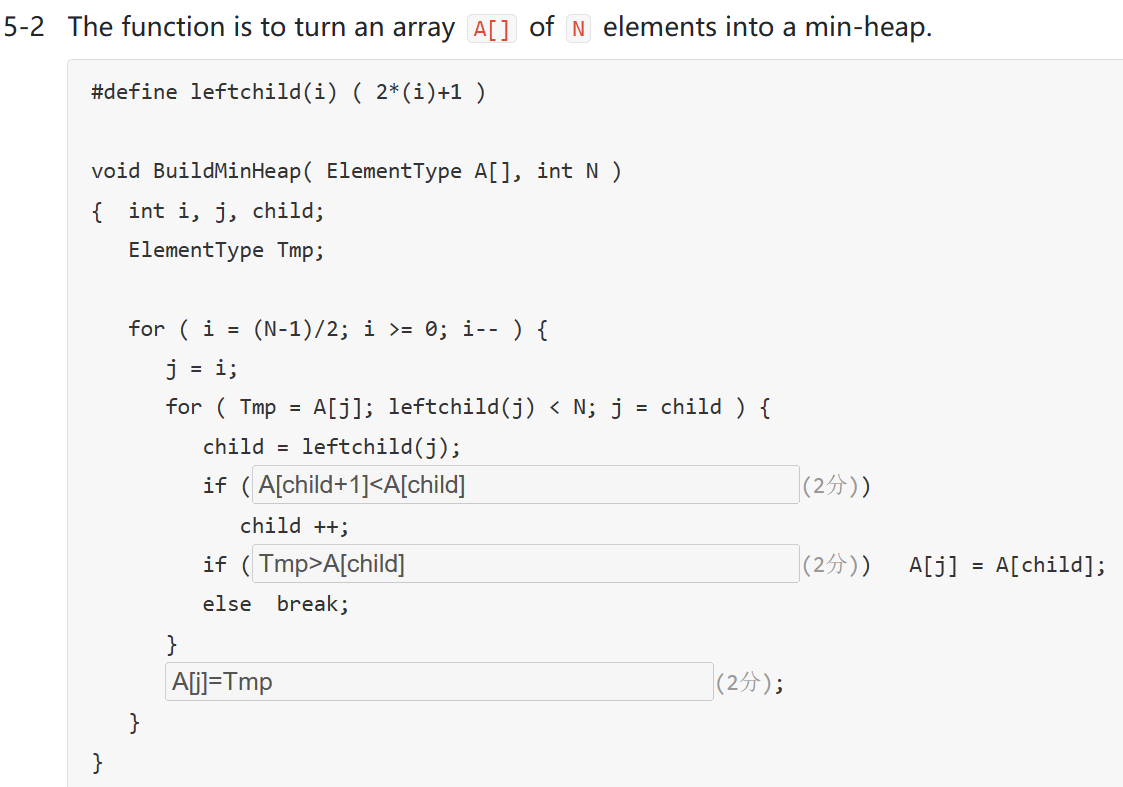
1. 除了右图的几条边还可能存在另外的边：0->2和0->4
2. 
3. For a directed graph, comparing to the representation of the adjacency lists, the adjacency matrix representation is easier to: (3分) A

A.find the in-degree of a vertex

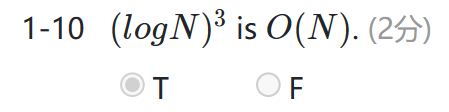
B.find the vertices that are adjacent from a vertex

C.do DFS

D.do BFS

1. Probing探测是要算进去没冲突的，如果问线性探测要几次则是冲突后的探测
2. 
3. If N numbers are stored in a singly linked list in increasing order, then the average time complexity for binary search is O(logN). (2分)

 F 链表不能用二分查找法

1. 
2. DFS详见PPT的09d第3页



1. 函数题6-1：17-18qm Shortest Path [2]
2. 函数题6-1：18-19qm Is Topological Order
3. During the sorting, processing every element which is not yet at its final position is called a "run". To sort a list of integers using quick sort, it may reduce the total number of recursions by processing the large partion first in each run.

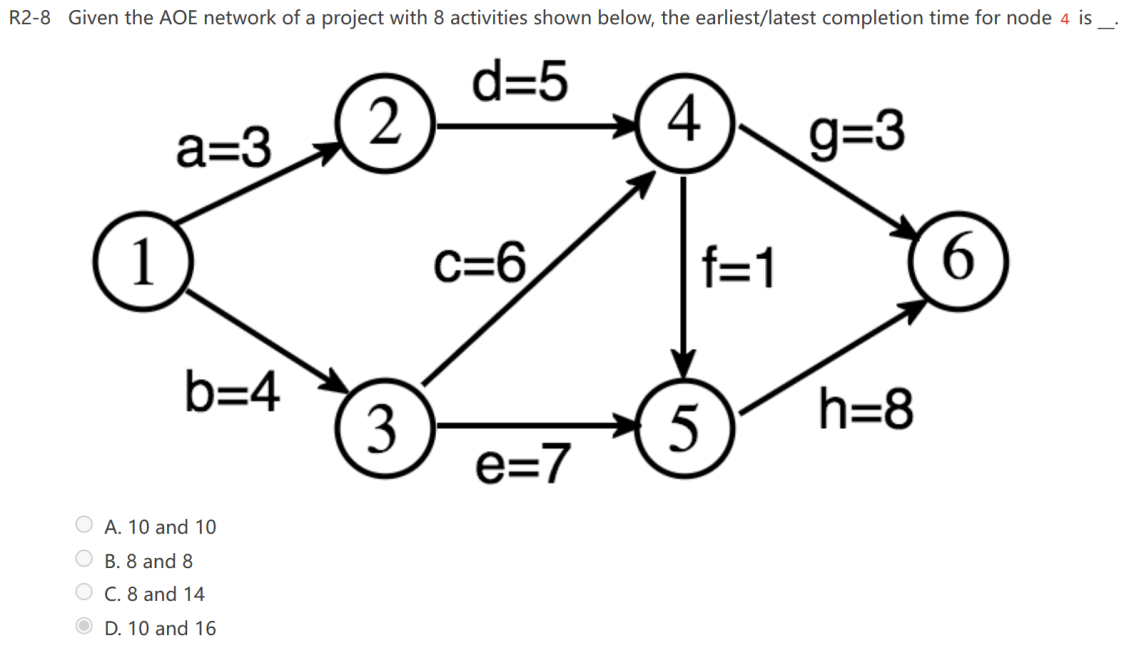
F 然而，处理较大划分先于较小划分并不能在每次划分中减少总递归次数。快速排序的递归次数取决于划分的平衡性,如果能将数组均匀地划分为两部分，递归深度会相对较小。

1. During finding all articulation points in a connected graph, if v1 is the parent of v2 in the depth-first tree, Num(v1) must be less than Num(v2).
2. The time complexity of traversing a binary tree with N nodes and height H in levelorder (with the help of with an auxiliary queue) is \_\_.O(N)

在每一层遍历时，将当前节点的左右子节点（如果存在）加入队列，然后按照队列的顺序依次访问节点。即使在不满二叉树的情况下，我们仍然需要访问所有的节点，因此时间复杂度仍然是O(N)。

The space complexity of iteratively traversing a binary tree with N nodes and height H in inorder (with the help of an auxiliary stack) is \_\_O(H)

在迭代中序遍历中，我们使用一个栈来辅助遍历二叉树。在遍历的过程中，栈中最多会保存H个节点，其中H是二叉树的高度。这是因为在中序遍历中，需要先遍历左子树，而左子树的深度最大为H

A

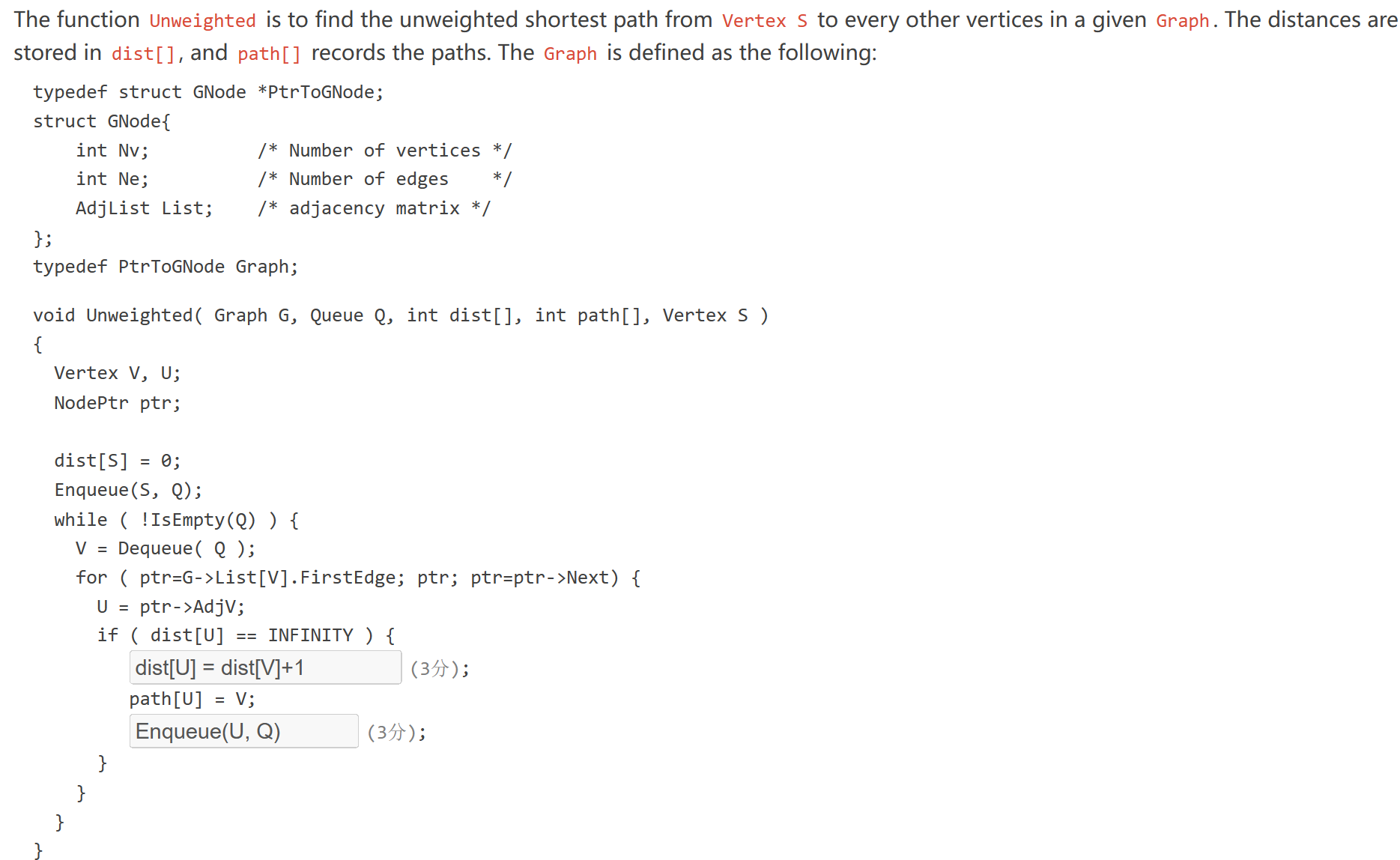
1. Graph G is an undirected completed graph of 20 nodes. Is there an Euler circuit in G? If not, in order to have an Euler circuit, what is the minimum number of edges which should be removed from G? C

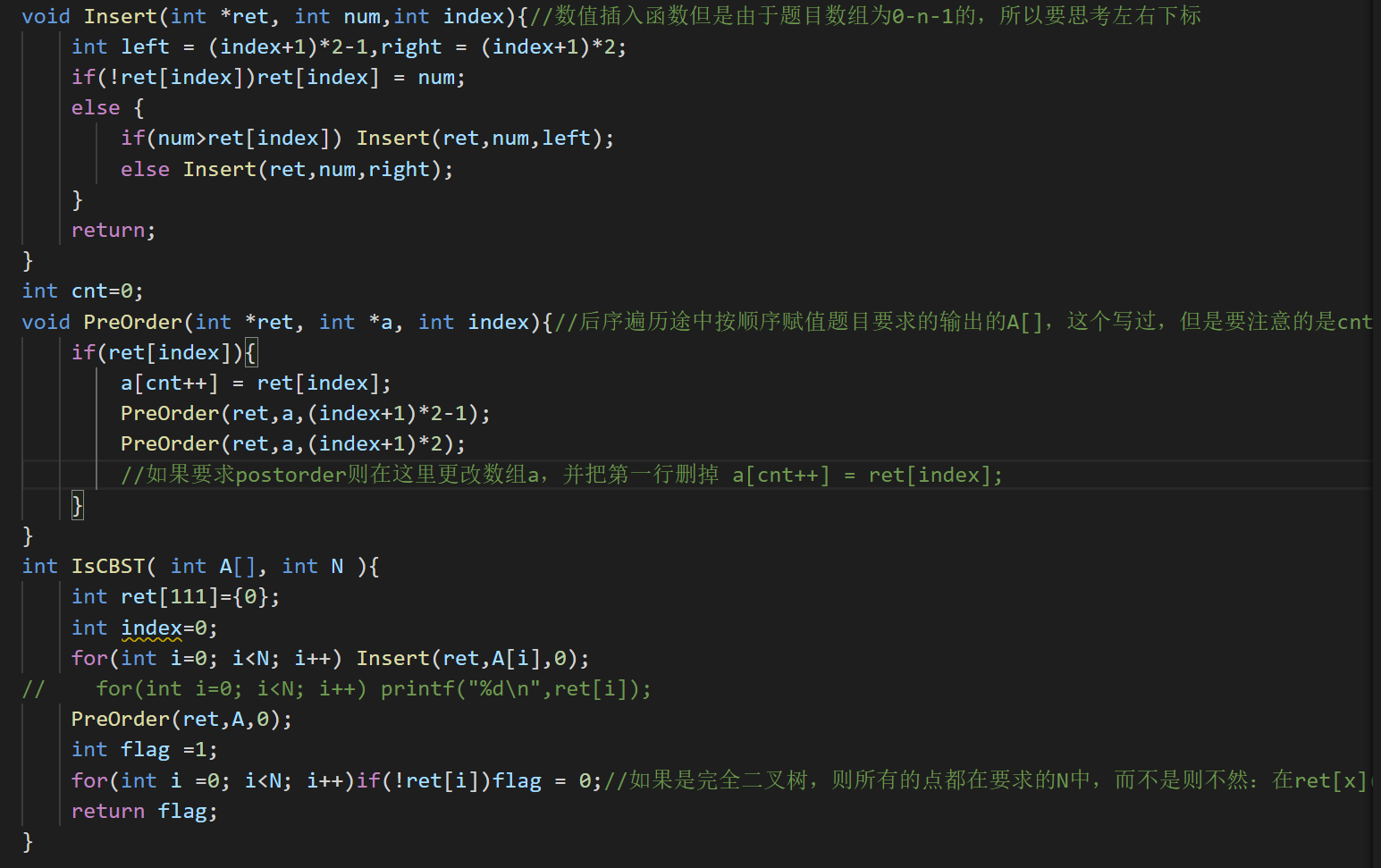
A.No, Graph G has no Euler circuit. 20 edges should be removed.

B.No, Graph G has no Euler circuit. 40 edges should be removed.

C.No, Graph G has no Euler circuit. 10 edges should be removed.

D.Yes, Graph G has an Euler circuit



44. 函数题6-1:19-20qm IsCBST

1. In hashing, when the loading density approaches 1, the operarion INSERTION will be seriously slowed down if the separate chaining method is used to solve collisions.

F 装载因子是指哈希表中已存储元素的数量与哈希表大小的比值。当装载因子接近1时，说明哈希表中的槽位（buckets）几乎全部被占用，每个槽位中链表的长度可能会很长。

In hashig with open addressing method，rehashing is definitely necessary when \_\_.

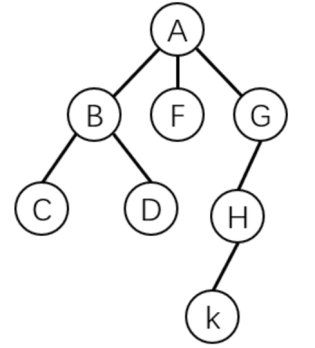
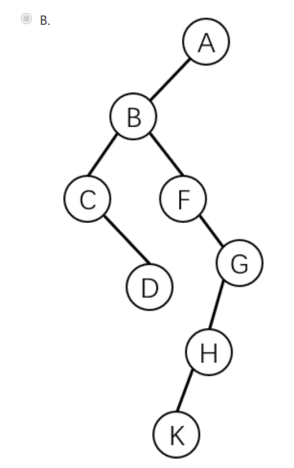
1. 在使用开放寻址方法的散列中，当\_\_时，重新散列是绝对必要的。 A

A.an insertion fails插入失败

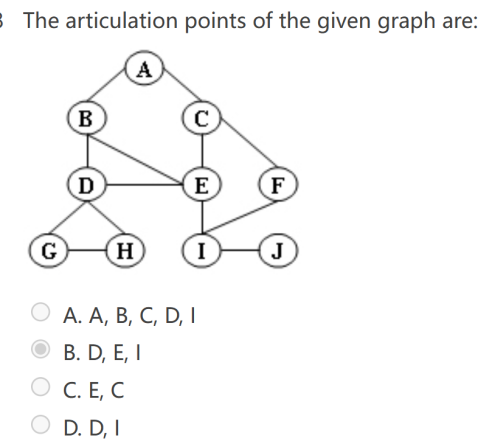
B.the hash table is half full哈希表满了一半

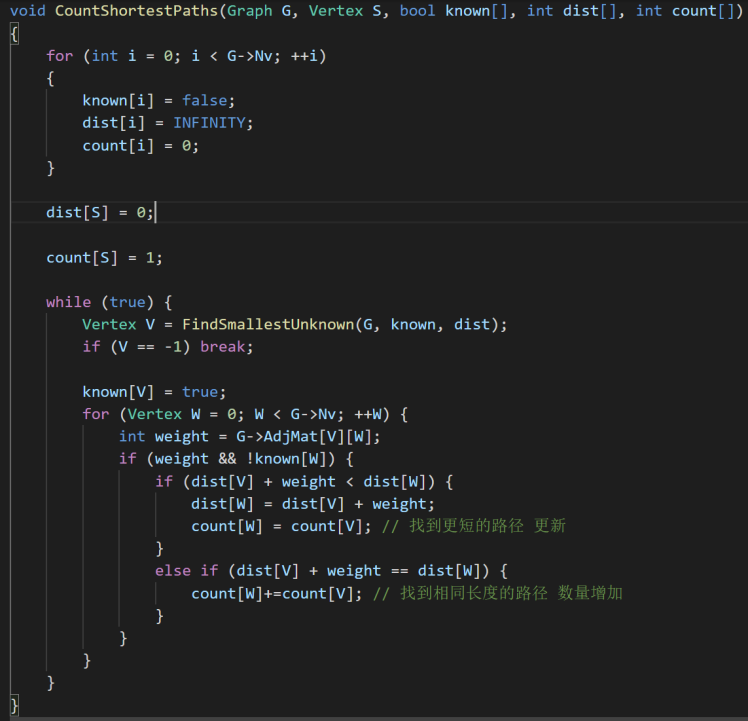
C.primary clustering occurs发生主要群集

D.the hash function is not uniform哈希函数不统一

1. The following figure shows a tree. Which one is its corresponding binary tree with the "first child/next sibling" representation?
2. An inversion in an array A[ ] is any ordered pair (i,j) having the property that i<j but A[i]>A[j]. Given array A: {3，87，12，61，70，26，45}，after the first partition of Quicksort with Median3 pivot selection, the number of inversions will be decreased by \_\_\_6\_\_.

这里有两个考点，第一个是逆序对的数量，所有顺序反的都算，所以初始的逆序对有5+2+2=9个；第二个考点是medium3算法，将首位中间排序后记得要将right-1和center的值交换，然后将right-1的值作为pivot，最后一次快排后的顺序是3 26 12 45 70 87 61

1. The articulation points of the given graph are:就是关节点，删掉了图就变得不连通点
2. D
3. The function CountShortestPaths is to find the number of shortest paths from Vertex S to every other vertices in a given graph (positive weights only). The distances and the numbers of shortest paths are stored in dist[] and count[], respectively. The Graph is defined as follows:

typedef struct GNode \*PtrToGNode;

struct GNode{

int Nv; /\* Number of vertices \*/

int Ne; /\* Number of edges \*/

int AdjMat[MaxVertexNum][MaxVertexNum]; /\* adjacency matrix \*/

};

typedef PtrToGNode Graph;

1. Given a min-heap with unique keys, there are at least h-1 keys less than any key in level h (the root level is 1).

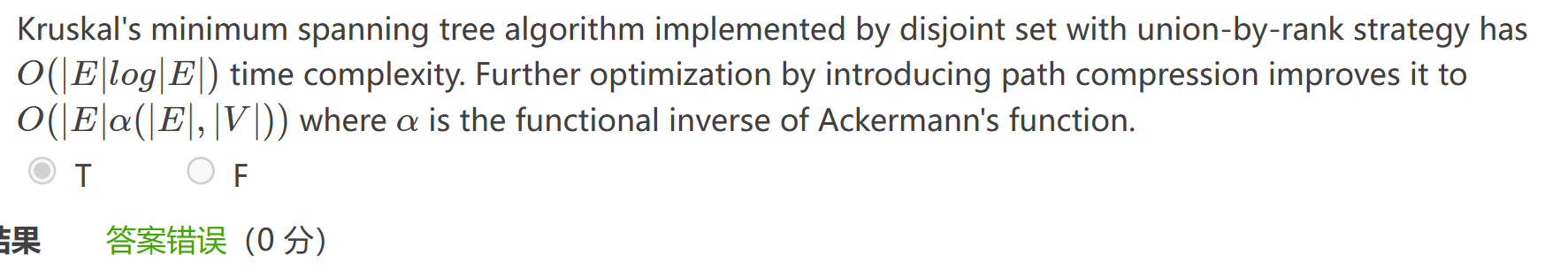
T

1. An inorder threaded binary tree cannot make the postorder traversal more efficient.

T

1. For binary heaps with N elements, the BuildHeap function (which adjust an array of elements into a heap in linear time) does at most N−log(N+1) comparisons between elements.

F 最坏的情况是max变成min先上寻到根结点然后在下沉到叶子结点，这里要注意比较的时候是比较两次找到两个子节点中教小的那个所以答案是两倍N−log(N+1)

1. Which sorting method can find the final position of at least one element within the sorted list after each run?

1.selection sort 2.quick sort 3.Shell sort 4.merge sort 5.heap sort

1. Which of the following is the sufficient and necessary condition that an undirected connected graph has Euler circuit? D

A.At most 1 vertex has an odd degree

B.Exactly 2 vertices have odd degrees

C.At most 2 vertices have odd degrees

D.All the vertices have even degrees

欧拉路径path和欧拉回路circuit是不一样的，前者最多两个奇数度数。起始点分别是这两个奇数度数的点，后者全部都是偶数度数，要求起始点同一个。

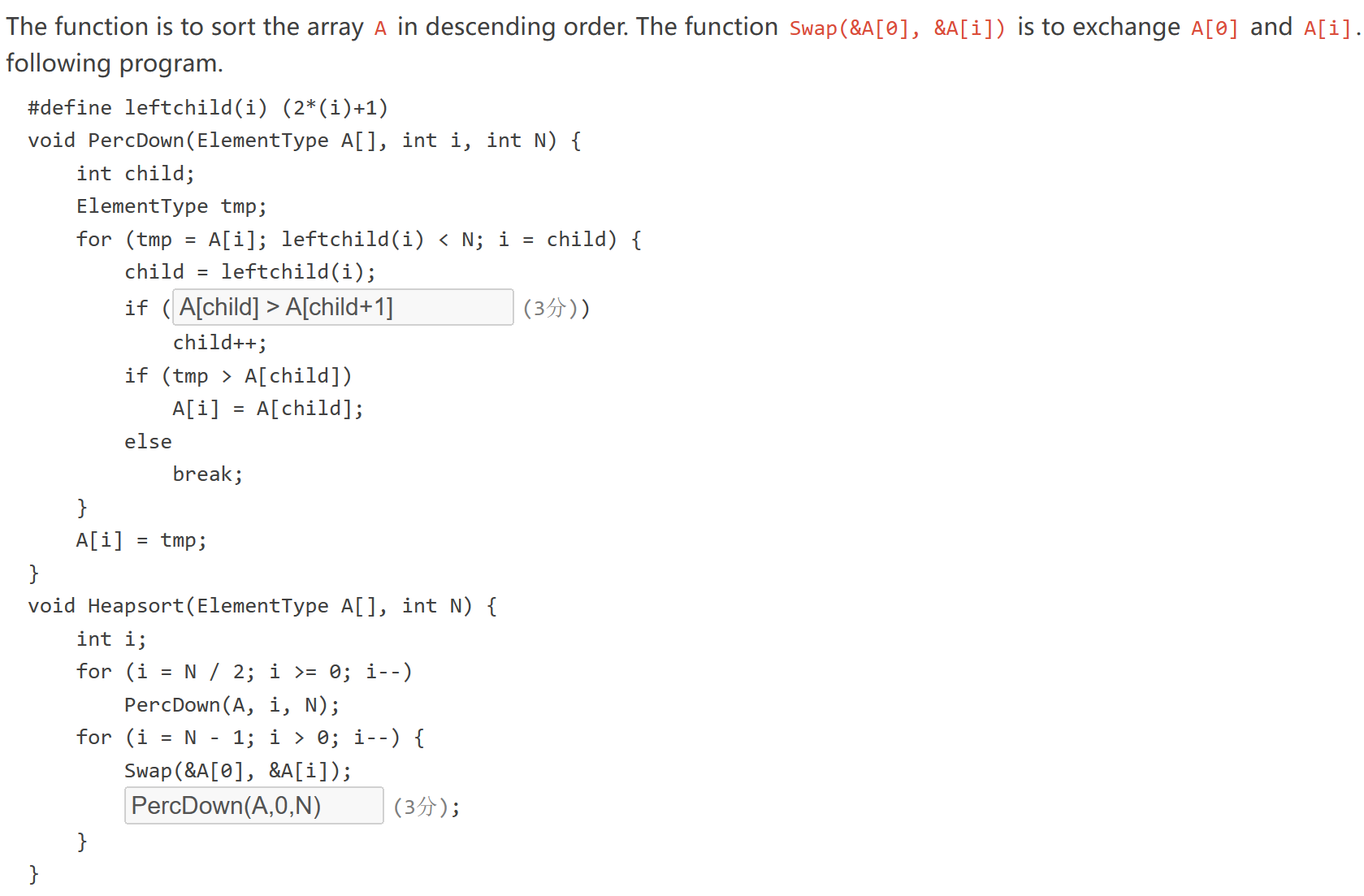
1. Let us convert a general tree T into a binary tree BT. Suppose that there are n0 leaf nodes in T and m0 leaf nodes in BT. Which of the following relationship between n0​and m0 is true? A

A.n0≥m0​

B.n0≤m0​

C.n0=m0​

D.cannot be determined



第1空：child + 1 < N && A[child + 1] < A[child]  
第2空：PercDown(A, 0, i)

这里显示用最小堆的方法建堆，然后deletemin放到堆的最后

