Questions 1 to 4 refer to the following algorithm.

Algorithm: P($a[l \ldots r]$)

Input: an array $a[l \ldots r]$ of real numbers

第1个 最后1个

begin

if $l = r$ **return** l

else

a的前半 放进去

$ll = P(a[l \ldots \lfloor (r+l)/2 \rfloor])$

后半

$rr = P(a[\lfloor (r+l/)2 \rfloor +1 \ldots r])$

if $a[ll] > a[rr]$ **return** ll

else **return** rr 若前半

end

1 2 3 4 5 6

1. Which algorithm design technique is employed in the above algorithm? **2.5**

[A] Brute Force technique
[B] Greedy technique
[C] Divide- and-Conquer
[D] Dynamic Programming
[E] Time and Space trade-off

2. For $n = 2^k$ and $k \geq 1$, the time complexity of the algorithm can be best expressed by

[A] $T(n) = T(n/2) + 1$ A
[B] $T(n) = T(n/2)$
[C] $T(n) = 2T(n/2) + 1$
[D] $T(n) = 2T(n/2)$
[E] None of the above

3. The time complexity of the algorithm is **2.5**

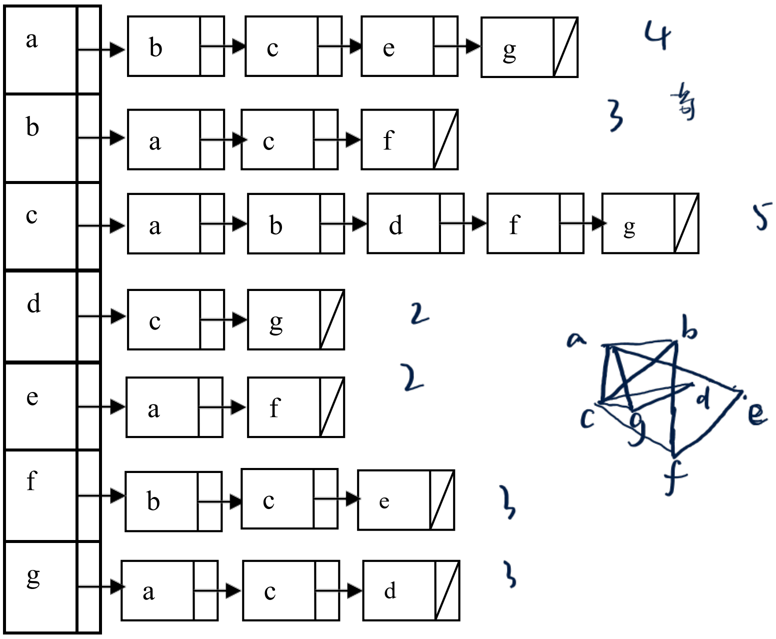
[A] $O(n)$ B
[B] $O(\log n)$
[C] $O(n^2)$
[D] $O(n \log n)$
[E] None of the above

4. What is the output of the algorithm for the input a[0..7] = [12, 12, 12, 12, 12, 12, 12, 12]? **2.5**

1 2 3 4 5 6 7 8

[A] 1
[B] 3 E
[C] 5
[D] 7
[E] 12

Questions 5 to 8 refer to the graph G represented by the following adjacency list



4

3 奇

5 奇

2

2

3

3

12+4+6
12+10
=22

5. Which of the following statement is true: The total degree of the graph G is **2.5**

所有顶点degree的和。

G

[A] The total degree of the graph G is 20 and it has a Euler circuit
[B] The total degree of the graph G is 18 and it has a no Euler circuit
[C] The total degree of the graph G is 11 and it has a Euler circuit
[D] The total degree of the graph G is 29 and it has a Euler circuit
[E] The total degree of the graph G is 22 and it has no Euler circuit

6. Starting at the vertex *a* and resolving ties by the vertex alphabetical order, traverse the graph by breadth-first-search (BFS). Then, the order of vertices visited is **2.5**

a, b c e g d f

C

~~[A]~~ *a, b, d, e, f, c, g*
[B] *a, b, c, d, e, f, g*
[C] *a, b, c, e, g, d, f*
[D] *a, b, g, e, f, c, d*
[E] *a, b, f, g, c, d, e*

7. Starting at the vertex *a* and resolving ties by the vertex alphabetical order, traverse the graph by depth-first-search (DFS). Then, the last vertex bei~~ng visit~~ed is **2.5**

a b f e c g d
1 2 3 2

D

[A] *g*
[B] *f*
[C] *c*
[D] *d*
[E] *e*

8. Starting at the vertex *a* and resolving ties by the vertex alphabetical order, traverse the graph by depth-first-search (DFS). Then, the 5th vertex being visited is 2.5

D

[A] *f*
[B] *g*
[C] *d*
[D] *c*
[E] *b*

spanning tree：含所有顶点

9. Let *T* be a tree constructed by Dijkstra's algorithm in the process of solving the single-source shortest path problem for a weighted connected graph G. 2.5

单源最短路径

I. *T* is a spanning tree of G
II. *T* is a minimum spanning tree of G
III. T is a binary tree

weight最小

Which one of the following is correct for every weighted connected graph？

A

[A] I is true, II and III are false.
[B] I and II and III are true.
[C] I and II and III are false.
[D] II is true but I and III are false.
[E] I and III are true but II is false.

Questions 10 to 11 refer to the following Bubble sort algorithm.

```
ALGORITHM Bubble Sort (A[0..n − 1])
//Sorts a given array by bubble sort
//Input: An array A[0..n − 1] of orderable elements
//Output: Array A[0..n − 1] sorted in ascending order
for i=0 to n − 2 do
for j = n-1 down to i+1 do
if A[j ]<A[j-1 ] swap A[j ] and A[j - 1]
```

对n个数排序，n-1次迭代
可以从小到大，也可以从大到小
轮次 一共n-1轮
从[i+1~n-1]
比较 交换

10. The number of swapping operations needed to sort the numbers A[0..5]=[6，1，2，3，4，5] in ascending order using the Bubble sort algorithm is **2.5**

需要多少次交换？

[A] 10
[B] 4
[C] 5
[D] 15
[E] 20

第1次 2次

6 1 2 3 4 5
1 6 2 3 4 5

6 1 2 3 4 5
1 2 3 4 5 6

Answer question 14 and 15 by running Floyd's Algorithm for the All-Pairs Shortest-Paths Problem for the weighted graph of question 14-17 (nodes a, b, c, d, e, f are numbered as 1, 2, 3, 4, 5, 6)	
14. What is the length of the shortest path between vertices a and d with intermediate vertices numbered not higher than 2, i.e., at most contains vertex a and b? [A] 3 [B] 5 [C] ∞ [D] 7 [E] 9 1 a → b → d c e f	2.5
15. What is the length of the shortest path between vertices c and d with intermediate vertices numbered not higher than 3, i.e., at most contains vertices a, b and c? [A] 3 [B] 6 [C] ∞ [D] 11 [E] 9 c → a → b → d	2.5

16. For the statements below,	2.5

I. A problem in the class P can be solved in worst-case by a polynomial time algorithm.

II. A problem in the class NP can be solved by a non-polynomial time algorithms ↘错

III. A problem in the class NP can be verified in polynomial time

IV. Finding minimum spanning tree (MST) in a weighted undirected graph is an P Problem

V. 0/1 Knapsack problem is an NP-Complete Problem

Which one of the following is correct? II false

(E)

[A] I, IV and V are true, II and III are false

[B] I, II, IV and V are true but III is false

[C] I and II are false but III, IV and V are true

[D] II, IV and V are true but I and III is false

[E] None of the above

17. Assume that all text and patterns consists of letters in A, B, C, D. The value of ~~T in the~~ shift table built by the Horspool's Algorithm for the pattern DCCDADDC is	2.5

1 2 3 4 5 6 7 8

C

[A] t(A) = 4, t(B)=9, t(C)=6, t(D) =4

[B] t(A) = 4, t(B)=9, t(C)=1, t(D) =2

[C] t(A) = 3, t(B)=8, t(C)=5, t(D) =1

[D] t(A) = 4, t(B)=8, t(C)=1, t(D) =2

[E] t(A) = 4, t(B)=9, t(C)=1, t(D) =1

D = 1

A = 3

C 5

Question 1(27 marks)	
Consider the following problem. Given an array A consisting of n distinct integers A[1], … A[n]. It is known that there is a position p ($1 \le p \le n$), such that A[1], .., A[p] is in increasing order and A[p], A[p+1], …,A[n] is in decreasing order.	
1. Write a brute force algorithm to find the position p. What is the time complexity of your algorithm?	**5**
2. Devise a “divide and conquer" algorithm to find the position p.	**8**
3. Set up a recurrence relation for the number of comparisons made by your algorithm and explain it.	**7**
4. Based on the recurrence relation, show the complexity of your algorithm in big-O notation and prove it using either the iterative method or the substitution method, i.e., Mathematical Induction (for simplicity, you can assume that $n = 2^k$).	**7**
Question II (13 marks)	
1. Briefly describe the idea of the polynomial time reduction. Explain how to use it to prove a problem is NP-complete. NP → hard NP	**5**
2. 4-SAT Problem: for a Boolean formula in CNF in which each clause has exactly 4 literals, determine if there is an assignment of Boolean value to its variables so that the formula evaluates to true? (i.e., the formula is satisfiable). Prove 4-SAT Problem is NP-Complete.	**8**