c

1. Two main measures for the efficiency of an algorithm are **2.5**

   a) Processor and memory
   b) Complexity and capacity
   c) Time and space
   d) Data and space
   e) Data and memory

b.

2. Considering the following algorithm **2.5**

```
input m
count = 0
x = 1
while x < m do
begin
    x = x * 2
    count = count + 1
end
output count
```

   What is the output of the algorithm for m=~~2k+1~~ 2^k+1?

   a) m
   b) k+1
   c) k-1
   d) k
   e) 2k

b.

3. The time factor when determining the efficiency of an algorithm is measured by **2.5**

   a) Counting the microseconds
   b) Counting the number of key operations
   c) Counting the number of statements
   d) Counting the kilobytes of the algorithm
   e) Counting the number of variables

a

4. The space factor when determining the efficiency of an algorithm is measured by **2.5**

   a) Counting the memory needed by the algorithm
   b) Counting the number of statements
   c) Counting the kilobytes of the algorithm
   d) Counting the maximum disk space needed by the algorithm
   e) Counting the number of key operations

b.

5. Running time T(*n*), where '*n*' is input size of a recursive algorithm is given as follows: **2.5**

Which of the following algorithms, its time complexity can be expressed by the T(n)? a) Binary Search b) Merge Sort ~~c)~~ Insertion Sort ~~d)~~ Selection Sort ~~e)~~ Dijkstra's algorithm	
6. Considering the running time $T(n)$ given above in the question 5. Order of magnitude of $T(n)$ is ~~a)~~ O(n^2) ~~b)~~ O(n) c) O(n log n) ~~d)~~ O(n^3) ~~e)~~ O(n^n)	**2.5**
7. The worst case occurs in a linear search algorithm when a) Item is somewhere in the middle of the array b) Item is not in the array at all c) Item is the last element in the array d) Item is the last element in the array or is not there at all e) None of the above	**2.5**
8. The best case occurs in a linear search algorithm when a) Item is the first element in the array b) Item is not in the array at all c) Item is the last element in the array d) Item is not in the array at all e) None of the above	**2.5**
9. The time complexity of a linear search algorithm is a) O(n) b) O(log n) c) O(n^2) d) O(n log n) e) O(1)	**2.5**
10. Running time T(n), where n is input size of a recursive algorithm is given as follows: $\mathrm{T}(n) = \begin{cases} 1 & if \quad n = 1 \\ T(n/2) + 1 & if \quad n > 1 \end{cases}$	**2.5**

c

d

a

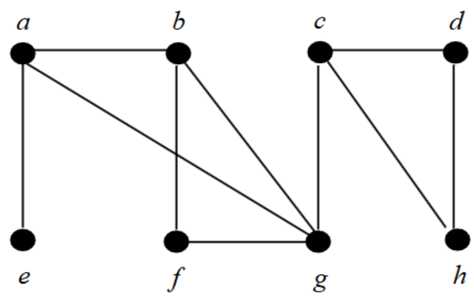
a

e

The time complexity of the algorithm is

a) $O(n^2)$
b) $O(n \log n)$
c) $O(n^3)$
d) $O(n^n)$
e) $O(\log n)$

Questions 11 to 13 refer to the following graph:



c

11. The total degree of the graph is **2.5**

a) 21
b) 18
c) 20
d) 19
e) 10

d

12. 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)~~ *a, b, d, e, f, c, h, g*
~~b)~~ *a, b, c, d, e, f, g, h*
~~c)~~ *a, b, f, g, c, d, h, e*
d) *a, b, e, g, f, c, d, h* √
e) None of the above

b

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

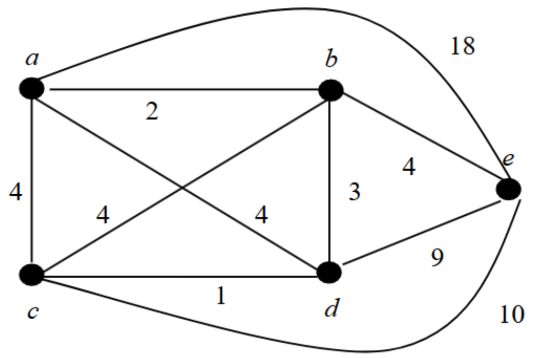
a) *h, g, f, e, d, c, b, a*

b) *a, b, f, g, c, d, h, e*	
c) *a, b, e, g, f, c, d, h*	
d) *a, b, d, c, e, f, g, h*	
e) None of the above	
a 14. 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**
I. T is a spanning tree of G ✓	
II. T is a minimum spanning tree of G X	
III. T is a binary tree X	
Which one of the following is correct?	
a) I is true, II and III are false ✓	
b) I and II and III are true X	
c) I and II and III are false X	
d) II and III are true but I is false X	
e) None of the above	
e 15. Let G be a weighted connected graph	**2.5**
I. If e is a minimum-weight edge in G, it must be contained in a MST. ✓	
II. If e is a minimum-weight edge in G, it must be contained in each MST. X	
III. If e is a maximum-weight edge in G, it must not be contained in any MST. X	
Which one of the following is correct?	
a) I and III are true, II is false	
b) I and II and III are true	
c) I and II and III are false	
d) II and III are correct but I is false	
e) None of the above	
16. Let G be a weighted connected graph	**2.5**
I. If the edge weights are all different G must have exactly one MST ✓	
II. If the edge weights are not all different G must have more than one MST X	

Which one of the following is correct?

a) I and III are true, II is false
b) I and II and III are true
c) I and II and III are false
d) II and III are true but I is false
e) None of the above

Questions 17 to 19 refer to the following weighted connected graph graph:



b

17. Let T be a minimum spanning tree of the graph computed using Kruskal's algorithm. The order of edges selected by Kruskal's algorithm is **2.5**

a) *(c,d) (a,b) (b,d) (a,c)*
b) *(c,d) (a,b) (b,d) (b,e)*
c) *(c,d) (a,b) (b,d) (a,d)*
d) *(c,d) (a,b) (b,d) (b,c)*
e) None of the above

d

18. Let T be a minimum spanning tree of the graph computed using Prim's algorithm. Assume vertex a is selected first, then the order of vertices selected by Prim's algorithm is **2.5**

a) *a, b, d, e, d*
~~b) *a, c, d, b, e*~~
~~c) *a, d, c, e, b*~~
d) *a, b, d, c, e*
e) None of the above

c

19. Assume the source vertex is a. Running Dijkstra's algorithm for the graph, after the termination, the labels for vertices are **2.5**

~~a) *a(0,-), b(2, a), c(4,a), d(5,b), e(6,b)*~~
~~b) *a(0,-), b(2, a), c(4,a), d(5,c), e(6,b)*~~
c) *a(0,-), b(2, a), c(4,a), d(4,a), e(6,b)*

~~d)~~ *a(0,-), b(2, a), c(6,b), d(4,a), e(6,b)*
e) None of the above

C

20. Assume the source vertex is a. Running Dijkstra's algorithm for the graph, after the termination, which one of the followings could be an order of vertices selected by Dijkstra's algorithm? **2.5**

a) *a, b, e, c, d*
b) *a, b, d, e, c*
c) *a, b, d, c, e*
d) *a, b, e, d, c*
e) None of the above

Questions 21 to 24 refer to the following Selection sort algorithm.

ALGORITHM *SelectionSort*($A[0..n-1]$)
//Sorts a given array by selection 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**
    $min = i$
    for $j = i+1$ **to** $n-1$ **do**
      if $A[j] < A[min]$ $min = j$
    if $i < min$ **do**
      swap $A[i]$ and $A[min]$

C

21. The time complexity of the Selection sort algorithm is **2.5**

a) $O(n \log n)$
b) $O(2^n)$
c) $O(n^2)$
d) $O(2n)$
e) None of the above

d

22. The number of key comparisons needed to sort the numbers A[0..5]= [6, 5, 4, 3, 2, 1] in ascending order using the selection sort algorithm is **2.5**

a) 10
b) 4
c) 3
d) 15
e) 20

C

23. The number of swapping operations needed to sort the numbers A[0..5]= [6, 5, 4, 3, 2, 1] **2.5**

a) 10
b) 4
c) 3
d) 15
e) 20

C

24. To merge the following two sorted sequences into a single sorted sequence, using the Merge algorithm given in the lecture, **2.5**

(3, 8, 10, 11, 15, 21, 24) and (25, 35, 45, 55, 80, 90, 100).

the number of key comparisons needed is
a) 9
b) 5
c) 7
d) 6
e) None of the above

Questions 25 to 28 refer to the following Longest Common Subsequence problem.

Let c[i,j] be the length of the Longest Common Subsequence of Xi = x1, x2,…, xi and Yj= y1, y2,…,yj . Then c[i,j] can be recursively defined as following:

$$c[i,j]=\begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ c[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } x_i = y_j \\ \max\{c[i-1,j],c[i,j-1]\} & \text{if } i,j>0 \text{ and } x_i \neq y_j \end{cases}$$

The following is an incomplete table for the sequences of AATGTT and AGCT.

		A	A	T	G	T	T
	0	0	0	0	0	0	0
A	0	1	1	1	1	1	1
G	0	1	1	1	2	2	2
C	0	1	1	1	2	2	2
T	0	1	1	2	2	3	3

b.

25. The value of c[3, 4] is
a) 1
b) 2
c) 3
d) 4
e) 5

C

26. The length of the longest common subsequence of AATGTT and AGCT is

a) 1
b) 2
c) 3
d) 4
e) 5

e

27. The longest common subsequence of AATGTT and AGCT is

a) AGCT
b) ATGT
c) AATG
d) AGC
e) AGT

b

28. The length of the longest common subsequence of AATGT and AGC is

a) 1
b) 2
c) 3
d) 4
e) 5

PART II (30 Marks)

Question 1 (18 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≤ p ≤n), such that A[1], .., A[p] is in increasing order and A[p], A[p+1], …, A[n] is in decreasing order.

1. Devise a “divide and conquer" algorithm to find the position p. **6**

2. Set up a recurrence relation for the number of key comparisons made by your algorithm and explain it. **6**

3. 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 = 2k). **6**

Question 2 (12 marks)

Given the following instance of the 0/1 Knapsack problem

item	weight	value
1	2	$12
2	1	$10
3	3	$20

The Knapsack Capacity W=3

Let V[i, j] be the value of the most valuable subset of the first i items that fit into the Knapsack of capacity j. Then V[i, j] can be recursively defined as following:

$$V[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ \max\{V[i-1,j],\quad v_i + V[i-1, j-w_i]\} & \text{if } j - w_i \geq 0 \\ V[i-1,j] & \text{if } j - w_i < 0 \end{cases}$$

1. Using dynamic programming, fill the missed values (filled with question marks) in the following table. **6**

capacity j

Item	i	0	1	2	3
	0	0	0	0	? 0
w_1=2, v_1=12	1	0	0	12	12
w_2=1, v_2=10	2	0	10	? 12	22
w_3=3, v_3=20	3	0	10	12	? 22

2. What is the value of the most valuable subset? **2**

22

3. Give an optimal subset of the instance based on the table. **2**

Put item 1 and item 2

4. What is the value of the most valuable subset if the capacity of the knapsack is 2? **2**

12

Q1. 1. Algorithm: Divide_and_conquer (A[1⋯n], l, r)

// Input: A[1⋯n], l, n

// Output: the position p

if l = r then
    return l
left = l
right = r
mid = $\left\lfloor \frac{l+r}{2} \right\rfloor$
if A[mid] < A[mid+1] then
    return Divide_and_conquer (A[1⋯n], mid+1, right)

else
    return Divide_and_conquer (A[1⋯n], left, mid)

2. $T(n) = \begin{cases} O(1) & n = 1 \\ T(\frac{n}{2}) + O(1) & n > 1 \end{cases}$

3. Suppose $n = 2^k$, then $k = \log_2 n$

$T(n) = T(\frac{n}{2}) + 1 = T(\frac{n}{2^2}) + 1 + 1 = \cdots = T(\frac{n}{2^k}) + 1 \cdot k = T(1) + k = \log_2 n + 1$
$= O(\log n)$