Part 1: Multiple Choice
Enter your answers on the Scantron sheet.

We will not mark answers that have been entered on this sheet.
Each multiple choice question is worth 2.5 marks.

In all the questions below, \( \log(x) \) means \( \log_2(x) \). You might find this fact useful:

\[
\log(xy) = \log x + \log y.
\]

1. Consider the following pairs of functions \( f(n), g(n) \). For which pair the functions are such that \( f(n) \) is \( O(g(n)) \) and \( g(n) \) is not \( O(f(n)) \)?
   (A) \( f(n) = n^2, g(n) = n \log n \)
   (B) \( f(n) = \log n, g(n) = \log(n^2) \)
   (C) \( f(n) = n, g(n) = 1000n \)
   (D) \( f(n) = 17, g(n) = \frac{1}{n} \)
   (E) \( f(n) = \log(n^3), g(n) = n \)

2. The values 10, 13, 3 are inserted, in the given order, in a hash table of size 7 and hash function \( h(k) = k \mod 7 \). Collisions are resolved using double hashing with secondary hash function \( h'(k) = 5 - (k \mod 5) \). In which entry of the table is the key 3 stored?
   (A) 2
   (B) 3
   (C) 4
   (D) 5
   (E) 6

3. Let \( T \) be an AVL tree. Suppose that after adding a new data item to \( T \), the tree becomes unbalanced. Let node \( z \) be the root of the smallest unbalance subtree in \( T \). Let \( y \) be the child of \( z \) with larger height, and let \( x \) be the child of \( y \) with larger height. After re-balancing the tree (and, regardless of whether an LL, LR, RR, or RL rotation was performed), which node will be the root of the subtree formerly rooted at \( z \)? Assume that all keys stored in \( T \) are different.
   (A) \( x \)
   (B) \( y \)
   (C) The node among \( x, y, z \) storing the largest key.
   (D) The node among \( x, y, z \) storing the middle key.
   (E) The node among \( x, y, z \) storing the smallest key.

4. How many different binary search trees containing the keys 1, 2, 3, 4 can be built such that a preorder traversal of the tree visits the node with key 1 first? Each key must appear in a tree once.
   (A) 3
   (B) 4
   (C) 5
   (D) 6
   (E) 7
5. Consider the following algorithm:

**Algorithm** $T(r)$
**Input:** Root $r$ of a proper binary tree.

if $r$ is a leaf then return 0
else {
    $p \leftarrow T(\text{left child of } r)$
    $q \leftarrow T(\text{right child of } r)$
    return $p + q + 1$
}

What does the algorithm compute?
(A) The number of nodes in the tree.
(B) The number of internal nodes in the tree.
(C) The number of leaves in the tree.
(D) The height of the tree.
(E) The number of descendants of $r$.

6. A database containing $10^9$ data items, each of size 10 bytes, is to be stored in disk. The disk hardware allows data blocks of $b$ bytes to be transferred to main memory every time that the disk is accessed. Which of the following data structures would allow the $\text{find}(\text{key})$ operation to be performed as efficiently as possible in the worst case?
(A) An AVL tree.
(B) A B-tree of order $b$.
(C) A B-tree in which every node stores $b$ data items.
(D) A B-tree in which every node has a size of $10^{10}/b$ bytes.
(E) A B-tree in which every node has a size of $b$ bytes.

7. Consider the following algorithm.

**Algorithm** $B(G, u)$
**Input:** Undirected, connected graph $G$, and vertex $u$ of $G$.
Mark($u$)
$c \leftarrow 0$
For each edge $(u, v)$ incident on $u$ do
    if $v$ is not marked then $c \leftarrow c + B(G, v) + 1$
return $c$

What does the algorithm compute?
(A) The number of edges of $G$.
(B) The number of vertices of $G$.
(C) The length of the path from $u$ to the farthest vertex $v$ of $G$.
(D) The number of connected components of $G$.
(E) The number of neighbours of $u$. 

3
8. Consider the following graph. Which edges, and in which order, are selected by Prim’s algorithm if it starts at vertex 1?

(A) (1,2), (2,7), (7,3), (3,6), (1,4), (3,5)
(B) (1,2), (2,7), (1,3), (3,6), (1,4), (3,5)
(C) (1,2), (3,7), (2,7), (3,6), (1,4), (3,5)
(D) (1,2), (2,7), (7,3), (1,4), (3,6), (3,5)
(E) (1,2), (3,7), (2,7), (1,3), (1,4), (3,5)

9. Consider the following weighted graph. Assume that we use Dijkstra’s algorithm to find shortest paths from vertex \(s\) to the other vertices in the graph. In which order are the final distance labels \(u.d\) computed? (Or in other words, in which order are the shortest paths computed?)

(A) \(s, 1, 2, 3, 4, 5\)
(B) \(s, 1, 4, 2, 5, 3\)
(C) \(s, 1, 4, 5, 2, 3\)
(D) \(s, 1, 4, 2, 3, 5\)
(E) \(s, 1, 4, 3, 2, 5\)

10. Which of the following statements is true for all possible depth first search traversals of the following graph starting at vertex 1?

(A) Vertex 2 must be the second vertex visited.
(B) Vertex 3 must be visited before vertex 6.
(C) Vertex 4 cannot be visited immediately before vertex 3.
(D) Vertex 6 cannot be visited before vertex 5.
(E) Vertex 7 cannot be the last vertex visited.
11. Let $G = (V, E)$ be an undirected, connected graph with $n$ vertices and $m$ edges. All vertices are initially un-marked. Consider the following algorithm:

**Algorithm `traverse(G, u)`**
**Input:** Undirected, connected graph $G$, and vertex $u$ of $G$.
Mark $u$
For each edge $(u, v)$ incident on $u$ do {
trim$(u, v)$
if $v$ is not marked then `traverse(G, v)`
}

Algorithm `trim(u, v)` performs $kmn$ operations, where $k$ is a constant value. Assume that $G$ is stored in an adjacency list. What is the time complexity of algorithm `traverse(G, u)` in the worst case?

- (A) $O(m + n)$
- (B) $O(mn)$
- (C) $O(mn^2)$
- (D) $O(mn(m + n))$
- (E) $O(n^2)$

12. Consider the following graph. Which one of the following IS NOT a valid ordering of the vertices if the graph is traversed using breadth first search (BFS)?

- (A) 1, 6, 3, 5, 4, 2
- (B) 4, 3, 2, 6, 5, 1
- (C) 2, 3, 4, 6, 5, 1
- (D) 3, 6, 4, 1, 2, 5
- (E) 5, 4, 3, 6, 1, 2

13. Which of the following is a valid topological ordering for the vertices of the following digraph.

- (A) g a d c i e b
- (B) g b e a c d i
- (C) g b e c a d i
- (D) g b a d j c i
- (E) g b e c d a i
14. Consider the following algorithm.

**Algorithm** Sort$(A, n)$

**Input:** Array $A$ containing $n$ different integer values.

**Out:** Array $A$ sorted in increasing order of value.

for $i \leftarrow n - 1$ downto 1 do {
    \[ m \leftarrow i \]
    \[ (*) \]
    \[ t \leftarrow A[m]; A[m] \leftarrow A[i]; A[i] \leftarrow t \]/ Swap $A[i]$ and $A[m]$
}

return $A$

Which of the following instructions must be inserted at the point marked (*) so that the algorithm correctly sorts the values stored in $A$ in increasing order of value?

(A) for $j \leftarrow 0$ to $i - 1$ do
    if $A[j] > A[m]$ then $m \leftarrow j$

(B) for $j \leftarrow i + 1$ to $n - 1$ do
    if $A[j] > A[m]$ then $m \leftarrow j$

(C) for $j \leftarrow 0$ to $n - 1$ do
    if $A[j] > A[m]$ then $m \leftarrow j$

(D) for $j \leftarrow i + 1$ to $n - 1$ do
    if $A[j] < A[m]$ then $m \leftarrow j$

(E) for $j \leftarrow 0$ to $n - 1$ do
    if $A[j] < A[m]$ then $m \leftarrow j$
Part 2: Written Answers
Write your answers directly on these sheets.
You might find this fact useful: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

15. (6 marks) Write a recurrence equation expressing the time complexity of the following algorithm. Explain your answer. A is a global array storing $n$ integer values. Assume that $n$ is a power of 2.

Algorithm $\text{rec}(n)$

Input: Value $n \geq 0$

if $n = 0$ then return 1
else {
    $c \leftarrow 0$
    For $i \leftarrow 0$ to $n - 1$ do $c \leftarrow c + A[i]$
    $c \leftarrow c + \text{rec}(n/2)$
    return $c$
}

16. (6 marks) Solve the following recurrence equation. Show all intermediate steps.

$T(1) = 0$
$T(n) = n - 1 + T(n - 1)$, for $n > 1$
For the following 2 questions you can use any of the algorithms studied in class. You do not have to write full descriptions of the algorithms. Just indicate which algorithm you would use, and which changes you need to make to the algorithm to answer each question. Indicate also how to pre-process the input for the algorithm. For example, given a road map with distances between cities, to find the shortest way of driving from city A to city B, your answer might be: Build a graph in which every node is a city and an edge represents a road. The length of an edge is the length of the corresponding road. Use Dijkstra’s algorithm to find the shortest path from A to B. This is the shortest route that should be taken.

17. (5 marks) There are \( n \) small islands in a lake and it is desired to build bridges to connect them so that each island can be reached from any other island via one or more bridges. The islands are numbered from 0 to \( n - 1 \). The cost of constructing a bridge between islands \( i \) and \( j \) is \( c(i, j) \). Describe an algorithm that determines which bridges must be built so that the total construction cost is minimum. For example, for the set of islands shown below, an optimum solution is the set of bridges in bold and it has total cost 16.

18. (5 marks) Consider a set of \( n \) cars parked in a small parking lot with a single exit. We wish to find the order in which all the cars can be moved out of the parking lot. We build a directed graph \( G = (V, E) \) in which every vertex corresponds to a car and there is an edge from vertex \( u \) to vertex \( v \) if car \( u \) is blocking car \( v \). Describe an algorithm that uses the information in \( G \) to determine the order in which the cars need to move out of the parking lot.
19. (12 marks) Let $G = (V,E)$ be an undirected graph in which every vertex is colored either black or white. Write an algorithm that given two black vertices $s$ and $t$ it decides whether there is at least one path $p$ from $s$ to $t$ that does not go through any white vertices. If such a path exists, the algorithm must return the value true, otherwise it must return the value false.

For example, for the following graph and $s = 0, t = 6$ the algorithm must return the value true (as paths $(0,1,6)$ and $(0,3,1,6)$ are as required); for $s = 1, t = 6$, the algorithm must also return true (as path $(1,6)$ only goes through black vertices); however, for $s = 0, t = 5$, the algorithm must return false (as the only paths from vertex 0 to 5 must go through white vertex 7).

20. (4 marks) Compute the time complexity of your algorithm for the previous question in the worst case, assuming that the graph is stored in an adjacency list. Explain how you computed the time complexity and give the order of the time complexity.
21. (12 marks) Consider two arrays $A$ and $B$, each storing $n$ integer values. All values in $A$ and $B$ are different. In both arrays the values are sorted in increasing order. Write an algorithm that merges $A$ and $B$ and stores their values in an array $C$ in such a way that the first $n$ values in $C$ are sorted in increasing order and the last $n$ values are sorted in decreasing order.

For example, if $A$ and $B$ are as follows:

\begin{align*}
A & \begin{bmatrix} 3 & 6 & 8 & 13 & 19 \end{bmatrix} \\
& \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \end{bmatrix} \\
B & \begin{bmatrix} 1 & 4 & 9 & 11 & 17 \end{bmatrix} \\
& \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \end{bmatrix}
\end{align*}

Then array $C$ must be this:

\begin{align*}
\begin{bmatrix} 1 & 3 & 4 & 6 & 8 & 19 & 17 & 13 & 11 & 9 \end{bmatrix} \\
& \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix}
\end{align*}

You cannot use any auxiliary data structures and you must write the complete algorithm (you cannot invoke algorithms for which you do not provide pseudocode).
22. (4 marks) Compute the time complexity of your algorithm for the previous question in the worst case. **Explain** how you computed the time complexity and give the order of the time complexity.

23. (5 marks) Insert the value 36 in the following (2,4)-tree. You **must** use the algorithm discussed in class for inserting information in a (2,4) tree. Show **all** intermediate trees.
24. (6 marks) Delete the value 5 from the following (2,4)-tree. You **must** use the algorithm discussed in class for removing information from a (2,4) tree. Show all intermediate trees.