1. Consider a LinkedList that has a `get(int i)` method to return the `i`-th element in the list. Write pseudocode to implement the method. Next, analyse (in order-notation, in terms of `n`) the following code:

```java
LinkedList L;
// ... LinkedList is initialized and now contains n items (code not shown).
for (int i=1; i<=n; i++) {
    sum = sum + L.get(i);
}
```

2. Consider this snippet of code:

```java
int[][] A = new int[n][n];
for (int i=0; i<n; i++) {
    if (i % 2 == 0) // i is a multiple of 2
        A[i] = new int[n];
    else
        A[i] = new int[1];
}
for (int i=0; i<A.length; i++)
    for (int j=0; j<A[i].length; j++)
        sum = sum + A[i][j];
```

It’s running time is:

(a) \(O(n \log n)\).
(b) \(O(n)\).
(c) \(O(n^2)\).
(d) \(O(n^2)\).

3. Suppose \(f(n) = 3n^3 + 2n^2 + n\). Consider these statements:

I  \(f(n) = O(n^3)\).
II  \(f(n) = O(n^4)\).
(a) Only I is true. 
(b) Only II is true. 
(c) I and II are both true. 
(d) I and II are both false.

4. The ratio of the running time of Bubble-Sort to that of Merge-Sort is:

(a) $O(n \log n)$. 
(b) $O\left(\frac{n}{\log n}\right)$. 
(c) $O\left(\log \frac{n^2}{\log n}\right)$. 
(d) $O(1)$.

5. Consider the following input with 10 keys (each is a letter) to QuickSort.

K G I K N V S S W Q

Assume the leftmost element is the partitioning element. Show the array after the partitioning step is complete.

6. Suppose the indices of the array that implements a binary heap start from 1. Now consider these assertions about finding the indices of the parent and left-child of the $i$-th element:

I $\text{parent}(i) = i/2$.  
II $\text{left}(i) = 2i + 1$.

(a) I and II are both true. 
(b) Only I is true. 
(c) Only II is true. 
(d) I and II are both false.

7. Delete 5 from this binary tree:

```
          10
         /  \
       5    12
      /  \   /  \
    2    7  6    8
```
8. Insert the bitstrings 100, 110, 011 and 010 into a simple trie.

9. Consider the statements
   I  In a simple trie, all the keys are at the leaves.
   II In a full trie, all the keys are at the leaves.
   (a) I and II are both true.
   (b) Only I is true.
   (c) Only II is true.
   (d) I and II are both false.

10. Draw an N DFA for the regular expression C(AB)*(A|C)

11. Suppose the height of a binary tree with one node is defined to be 1. A complete binary tree of height $h$ has
   (a) $2^h$ edges.
   (b) $2^h - 1$ edges.
   (c) $2^h - 2$ edges.
   (d) $2^{h+2}$ edges.

12. A ring is a connected undirected graph in which all the nodes form a ring (thus, each node has exactly two neighbors). The running time of Prim’s algorithm, using an adjacency list, on this graph is:
   (a) $O(V \log V)$
   (b) $O(V^2 \log V)$
   (c) $O(V^2)$
   (d) $O(V^3)$

13. Consider these two statements about a connected undirected graph with $V$ vertices and $E$ edges:
   I  $O(V) = O(E)$
   II $O(E) = O(V^2)$
   (a) I and II are both false.
   (b) Only I is true.
   (c) Only II is true.
   (d) I and II are both true.
14. Consider the following two statements about an undirected graph with $n$ vertices:

I. If every pair of vertices has an edge, there are exactly $\frac{n(n+1)}{2}$ edges.

II. If the graph has a spanning tree, the tree must have $\Theta(n)$ edges.

(a) I and II are both true.
(b) Only I is true.
(c) Only II is true.
(d) I and II are both false.

15. Suppose the shortest path from node $i$ to node $j$ goes through node $k$ and that the cost of the subpath from $i$ to $k$ is $D_{ik}$. Consider these two statements:

I. Every shortest path from $i$ to $j$ must go through $k$.

II. Every shortest path from $i$ to $k$ has cost $D_{ik}$.

(a) I and II are both true.
(b) Only I is true.
(c) Only II is true.
(d) I and II are both false.

16. Consider the contiguous load balancing problem where $s_i$ is the execution time of task $i$ and let $D_i^k$ be the optimal cost in assigning tasks $0, \ldots, i$ on to $k$ processors. Then, the dynamic programming recurrence can be stated as:

(a) $D_i^k = \max_j \min(D_{j-1}^k, \sum_{l=j+1}^{i} s_l)$.

(b) $D_i^k = \max_j \min(D_{j-1}^k, \sum_{l=j+1}^{i} s_l)$.

(c) $D_i^k = \max_j \max(D_{j-1}^k, \sum_{l=j+1}^{i} s_l)$.

(d) $D_i^k = \min_j \max(D_{j-1}^k, \sum_{l=j+1}^{i} s_l)$.

17. Consider the execution times of two algorithms I and II:

I. $O(n \log n)$

II. $O(\log(n^n))$

(a) Only I is polynomial.
(b) Only II is polynomial.
(c) I and II are both polynomial.
(d) Neither I nor II is polynomial.

18. For a travelling salesman problem with $n$ cities, the number of possible tours is:
19. Consider the following statements about NP-completeness:
   I  NP-completeness only applies to combinatorial optimization problems.
   II  An NP-complete problem cannot be solved in polynomial-time.

(a) Only I is true.
(b) Only II is true.
(c) I and II are both true.
(d) Neither I nor II is true.

20. Consider the following statements about NP-completeness:
   I  The traveling salesman problem is NP-complete.
   II  The minimum-spanning tree problem is not NP-complete.

(a) Only I is true.
(b) Only II is true.
(c) I and II are both true.
(d) Neither I nor II is true.

21. Consider the following statements about estimation:
   I  The sample mean is always larger than the sample variance.
   II  It is possible for the sample variance to be zero.

(a) Only I is true.
(b) Only II is true.
(c) I and II are both true.
(d) Neither I nor II is true.