THE UNIVERSITY OF AUCKLAND

SEMESTER TWO 2019
Campus: City

COMPUTER SCIENCE

Algorithms and Data Structures

(Time allowed: TWO hours)

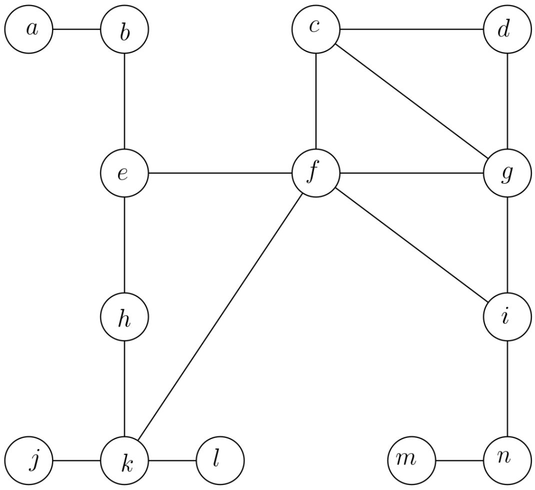
NOTE:

- Enter your full name, your student ID number and unique VERSION number on the top of the Teleform sheet.
- Sign your full name and write your ID number on the cover of this test script (see below).
- Keep this customized test script until your marks are finalized on Canvas. (On request, you may need to physically bring it to the course coordinator.)
- Shade in completely each answer on the Teleform sheet provided.
- All questions are worth the same number of marks.
- Use of any calculators or electronic devices is *not* permitted.
- There is no penalty for incorrect answers.

Student Signature: ____________________________________

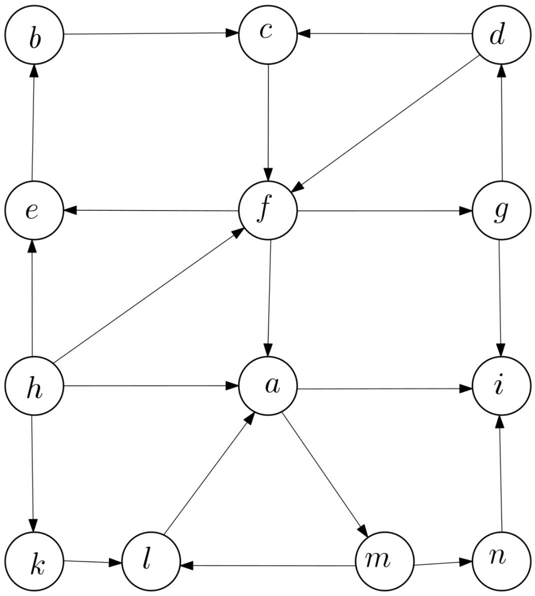
Student ID: ____________________________________

1. Consider the following graph. Do a PFS starting at vertex g. What is the 8th vertex visited (colored GREY). Here the priority of a vertex is its degree (lower number having higher prioity). Assume that if there is a choice you choose the vertex with lower alphabetical order.



A. k
B. e
C. h
D. b
E. m

2. Consider the following digraph. Do a BFS starting at vertex g. What is the 8th vertex visited (colored GREY). Assume that if there is a choice you choose the vertex with lower alphabetical order.



A. e
B. c
C. m
D. n
E. b

3. Which of the following statements is a FALSE statement about universal hashing.

A. With table size m, the probablilty of a collision is $1/m$ when picking a random hash function in a universal class of functions.
B. There are several ways to construct a universal hash family.
C. There is a set of hash functions and each application randomly picks one for each key.
D. There is a set of hash functions and each application randomly picks one and uses it until the application quits.
E. Over a sequence of k insert, lookup and delete operations (assuming hash table does not fill up), the expected total cost for all operations is $O(k)$.

4. Consider Dijkstra's, Bellman=Ford's and Floyd's shortest paths algorithms. How many of them are primarily based on a greedy (local-search) algorithm?

   A. 0
   B. 2
   C. 3
   D. 1

5. Suppose we want to find the 5th largest element of list $a[0..n-1]$. Which of the following is the most time-efficient way to do it?

   A. Sort the list with Quicksort and take the 5th element from the end.
   B. Sort the list with Mergesort and take the 5th element from the end.
   C. Use Quickselect to find the 5th order statistic.
   D. Sort the list with Insertion sort and take the 5th element from the beginning.
   E. Use Quickselect to find the (n-4)th order statistic.

6. An **elementary** operation in this question is only a comparison of two elements of an array. Find the number of **elementary** operations in the following piece of code.

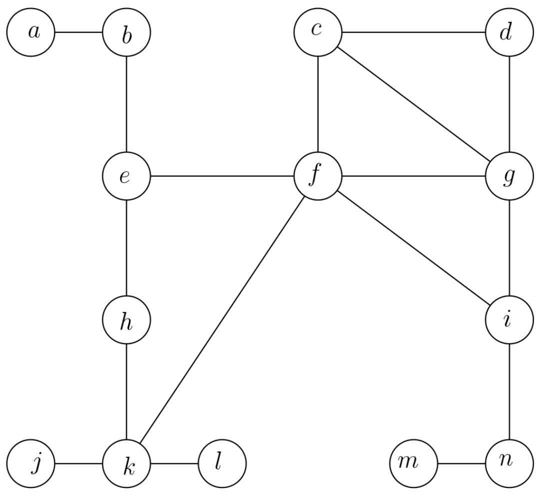
```
function BLAH(array a[0..n − 1])
    for i ← 1 to n do
        if i is a power of 2 then
            Do constant C number of elementary operations
            for j ← 1 to n do
                if j is divisible by 2 then
                    Do constant C number of elementary operations
    return something
```

   A. $C(\lfloor \log_2(n) \rfloor + 1)(1 + \lfloor \frac{n}{2} \rfloor)$
   B. $C^2 \lceil \log_2(n) \rceil \lceil \frac{n}{2} \rceil$
   C. $nC(\lfloor \log_2(n) \rfloor - 1)(1 + \lfloor \frac{n}{2} \rfloor)$
   D. $C \lfloor \log_2(n) \rfloor \lfloor \frac{n}{2} \rfloor$
   E. $C^2(\lfloor \log_2(n) \rfloor + 1) \lfloor \frac{n}{2} \rfloor$

7. Consider Dijkstra's, Bellman-Ford's and Floyd's shortest paths algorithms. How many of them are primiarily based on dynamic programming?

A. 2
B. 1
C. 3
D. 0

8. Consider the following graph. What is the eccentricity of vertex g?



A. 2
B. 6
C. 5
D. 3
E. 4

9. Which of the following statements describes the selection sort algorithm?

A. Build the binary heap by heapifying the array, remove the highest priority node until heap is empty.
B. Choose a pivot p. Partition the list: the left sublist with elements $\leq p$ including the pivot, the right sublist with elements $\geq p$. Recursively sort left and right sublists.
C. Choose a pivot p. Partition the list: the left sublist with elements $\leq p$, the pivot, the right sublist with elements $\geq p$. Recursively sort left and right sublists.
D. Split the input list into sorted and unsorted sublists. Repeat the following steps until the unsorted sublist is empty: choose the first element x in the unsorted sublist; find the last element $y \leq x$ in the sorted sublist; insert x after y in the sorted sublist.
E. Split the input list into sorted and unsorted sublists. Repeat the following steps until the unsorted sublist is empty: find a maximal element of the unsorted part by sequential scan; move the maximal element to the head of the sorted part.

10. Consider the algorithm below:

---

function INSERTINORDER(sorted array $a[0..n-1]$, **Value**)
  $i \leftarrow 0$
  while $a[i] <$**Value** and $i \leq n-1$ **do**
    $b[i] \leftarrow a[i]$
    $i \leftarrow i+1$
  $b[i] \leftarrow$ **Value**
  while $i \leq n-1$ **do**
    $b[i+1] \leftarrow a[i]$
    $i \leftarrow i+1$
  return b

The running time in **AVERAGE** case is $f(n)$. Which of the following statements is correct?

A. $f(n)$ is $\Theta(1)$
B. $f(n)$ is $\Theta(\log(n))$
C. All answers are incorrect.
D. $f(n)$ is $\Theta(n)$
E. $f(n)$ is $\Theta(n^2)$

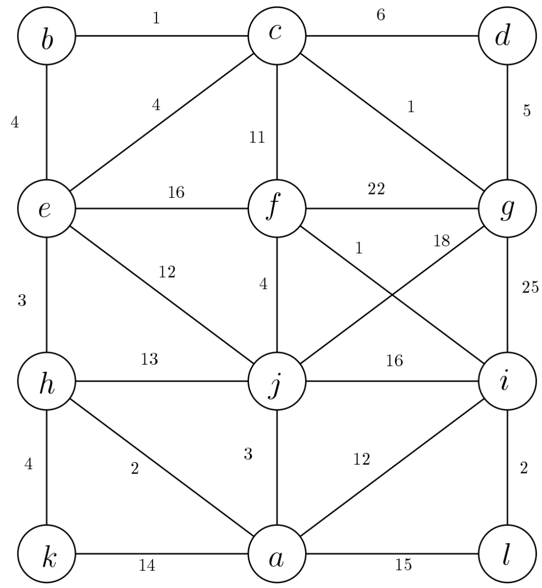
11. We are looking for 7 in $[1, 2, 3, 4, 11, 12, 0]$. Which of the following statements best describes the first steps of a binary search algorithm?

A. We cannot use binary search because the list is not sorted.
B. Find the middle element. It is 3. Because $7 > 3$ then look for 7 in $[1, 2]$.
C. Find the middle element. It is 3. Because $7 > 3$ then look for 7 in $[3, 4, 11, 12, 0]$.
D. Take 1 as the pivot. Partition the list into $[]$ and $[2, 3, 4, 11, 12]$. Because $7 > 1$ we recursively search for 7 in $[2, 3, 4, 11, 12, 0]$.
E. Find the middle element. It is 3. Because $7 > 3$ then look for 7 in $[4, 11, 12, 0]$.

12. Which of the following statements is **TRUE** about the insertion sort algorithm?

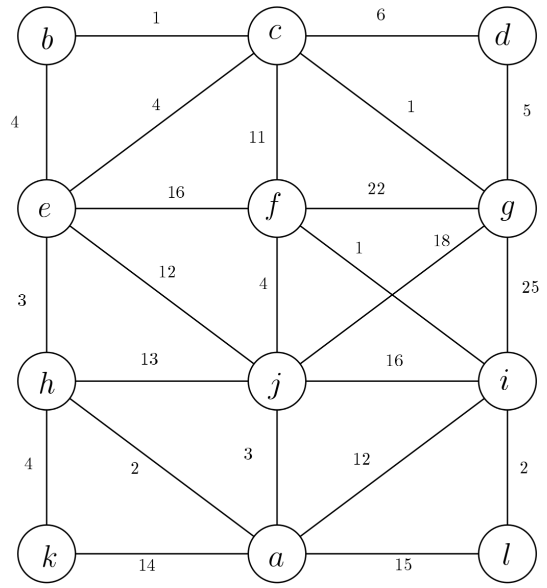
A. The worst time is $\Theta(n^2)$.
B. It is not stable.
C. It is not in-place.
D. The best time is $\Theta(n^2)$.
E. It is insensitive to the input data.

13. Consider the following edge-weighted graph. if we run Dijkstra's shortest paths algorithm starting at node c, what is the distance of a closest WHITE vertex (to c) after six of them have turned BLACK? Assume if we have a choice, we pick the smallest labeled vertex.



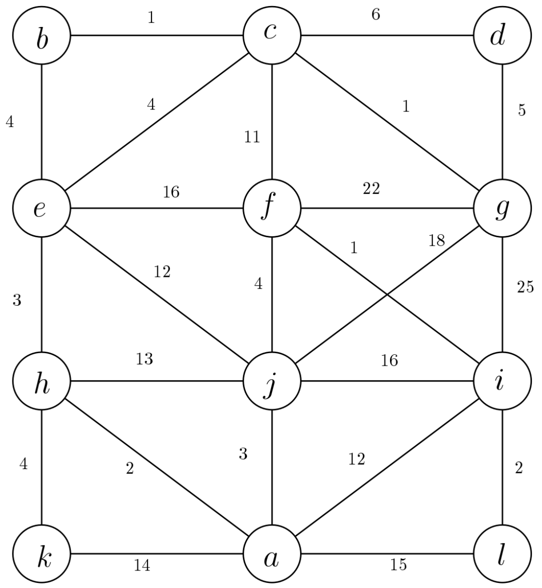
A. 9
B. 11
C. 15
D. 16
E. 19

14. Consider the following edge-weighted graph. If we run Dijkstra's shortest paths algorithm starting at node b, what is the sixth vertex colored BLACK (e.g., distance away from b is known)? Assume if we have a choice, we pick the smallest alphabetically labeled vertex.



A. j
B. h
C. k
D. f
E. d

15. For the following edge-weighted graph, how many unique minimum spanning trees are there?

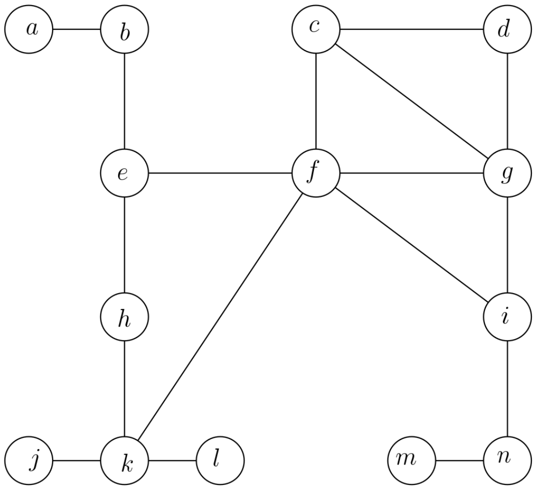


A. 3
B. 1
C. 5
D. 4
E. 2

16. We run the array implementation of heapsort. We have built a binary heap $[10, 9, 8, 7, 6, 5, 4, 2, 1]$. Which of the following arrays correspond to the next step of the algorithm?

A. Remove 10 and get $[9, 7, 8, 4, 6, 5, 1]$.
B. Remove 10 and get $[9, 7, 8, 2, 6, 5, 4, 1, 10]$.
C. Remove 10 and get $[9, 7, 8, 2, 6, 5, 4, 1]$.
D. Remove 1 and get $[10, 9, 8, 7, 6, 5, 4, 2, 1]$.
E. Remove 4 and get $[10, 9, 8, 7, 6, 5, 2, 1]$.

17. Consider the following graph. What is its diameter minus its radius?



A. 3
B. -1
C. 2
D. 0
E. 1

18. Which of the following problems is not NP-hard?

A. Finding a shortest path (between two vertices) in an edge-weighted graph with no negative length cycles.
B. Finding a shortest path (between two vertices) in an edge-weighted graph with all edge weights negative.
C. Finding a longest path (between two vertices) in an edge-weighted graph with only positive edge lengths.
D. Finding a longest path (between two vertices) in an edged-weighted graph with no negative length cycles.
E. Finding a longest path (between two vertices) in an unweighted graph (e.g. all edges have length 1).

19. Consider the algorithm below:

function REVERSE(array $a[0..n-1]$, the left pointer i, the right pointer j)
  if $i < j$ **then**
    SWAP(i, j, a)
    REVERSE($a, i+1, j-1$)
  else
    return a

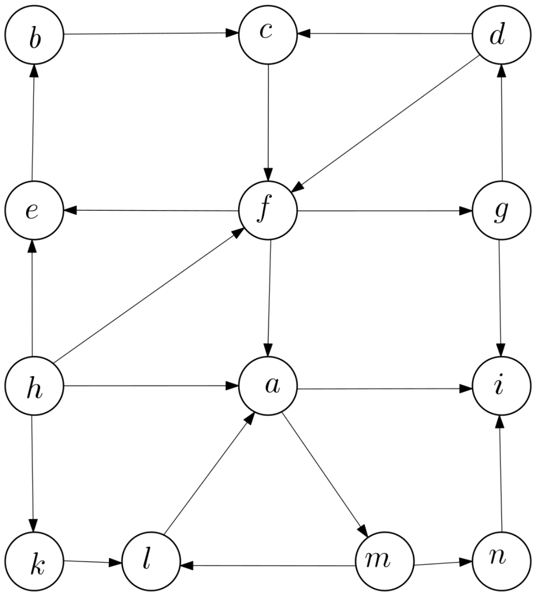
An **elementary** operation in this question is a comparison or a swap of two elements of the list. What is the recurrence relation for this algorithm?

A. $T(n) = T(n-1) + n, T(0) = 0$
B. $T(n) = T(n-2) + 2, T(0) = 7$
C. $T(n) = T(n-2) + 1, T(0) = T(1) = 0$
D. $T(n) = T(n-1) + 1, T(0) = 0$
E. $T(n) = T(n-2) + 1$

20. Which of the following statements is not a good property of a good hashing scheme with hash function h over a set of keys S with table size m.

A. The function h is fast to compute (polynomial in the size of a key $x \in S$, but constant time with respect to n).
B. h is a perfect hash function.
C. The hash locations of the keys S are spread out to minimize collisions (for lookup, insert, deletes,...).
D. The hash function maps keys to the first half of the table at indices $1, 2, \ldots, m/2$.
E. Use a relatively small table size $m = O(n)$ that does not affect performance.

21. Consider the induced sub-digraph obtained with nodes $\{a, c, f, h, i\}$ of the following digraph. How many different topological orders are there of these nodes?



A. 5
B. 4
C. 3
D. 6
E. 2

22. Let n be the order and m be the size of an edge-weighted graph G. If using a binary heap and adjacency lists to represent G, what is the running time of Dijkstra's shortest paths algorithm?

A. $O(m \log n)$
B. $O(m + n \log m)$
C. $O(n \log m)$
D. $O(n^2)$
E. $O(n \cdot m)$

23. The recurrence relation $T(n) = 4T(n-1) + 1, T(0) = 0$ has a solution which is in $\Theta(f(n))$. What is $f(n)$?

    A. $n \log_4(n)$
    B. n^2
    C. 4^n
    D. n^3
    E. n

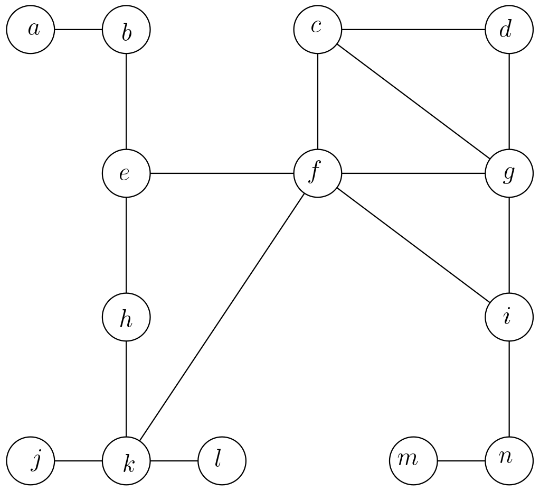
24. Which of the following statements about $f(n) = 10^{n^2} + (\log_{60}(n))^{10} + n^{1000}$ is FALSE?

    A. $f(n)$ is $\Theta(n^{1000})$
    B. $f(n)$ is $\Theta(10^{n^2})$
    C. $f(n)$ is $O(n^n)$
    D. $f(n)$ is $\Omega(\log_{60}(n))$
    E. $f(n)$ is $\Omega(n^2)$

25. Suppose we have a hash table of size 11 using the hash function $h(x) = x \bmod 11$. If we do linear probing to the right, what is the state of the hash table when adding $3, 13, 14, 22, 16, 27, 24$?

    A. [22,$\square$,3,13,14,16,27,24,$\square$,$\square$,$\square$]
    B. [22,14,13,3,27,16,$\square$,$\square$,$\square$,$\square$,2]
    C. [22,13,3,14,16,27,24,$\square$,$\square$,$\square$,$\square$]
    D. [22,$\square$,13,3,14,16,27,24,$\square$,$\square$,$\square$]
    E. [22,14,13,3,27,16,2,$\square$,$\square$,$\square$,$\square$]

26. Consider the following graph. Do a DFS starting at vertex g. What is the 8th vertex visited (colored GREY). Assume that if there is a choice you choose the vertex with lower alphabetical order.



A. k
B. n
C. e
D. b
E. h

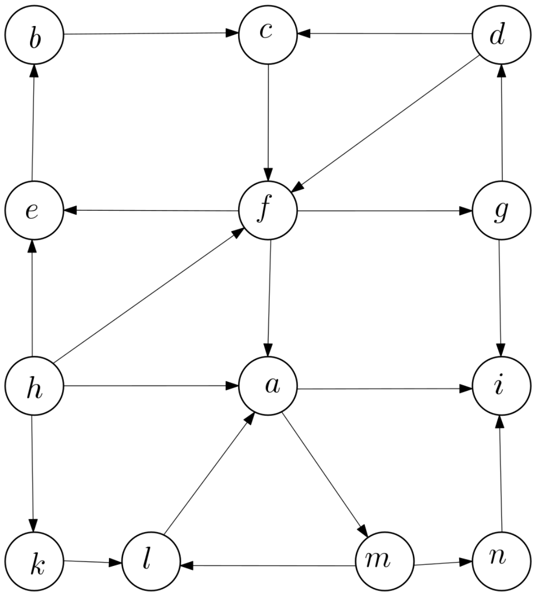
27. Consider Bellman-Ford's algorithm for computing single-source shortest paths. Which of the following statements is FALSE?

A. The algoritm runs in time $\Omega(nm)$ for adjacency lists.
B. For sparse graphs the running time is the same when using either adjacency matrix or adjacency lists.
C. For dense graphs the running time is the same when using either adjacency matrix or adjacency lists.
D. The algoritm runs in time $\Theta(nm)$ for adjacency lists.
E. The algoritm runs in time $O(nm)$ for adjacency lists.

28. Which of the following minimum spanning tree algorithms benefits from using the **union-find** ADT?

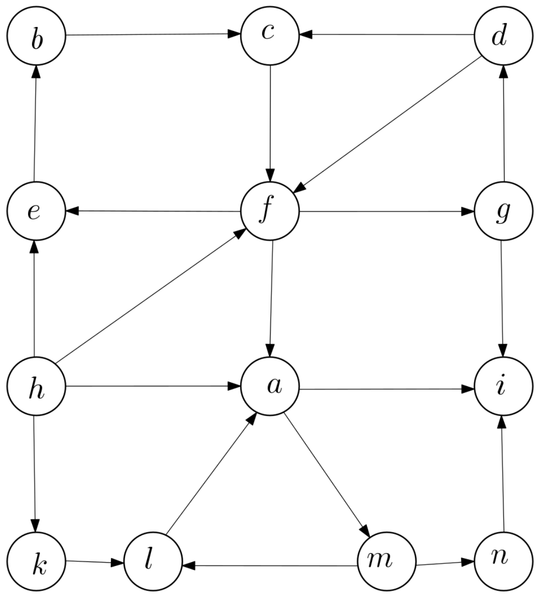
A. Boruvka's
B. Kruskal's and Prim's
C. Kruskal's
D. Boruvka's and Prim's
E. Prim's

29. Consider the following digraph. How many strongly connected components are there?



A. 2
B. 5
C. 3
D. 4
E. 6

30. Consider the following digraph. What is its girth?



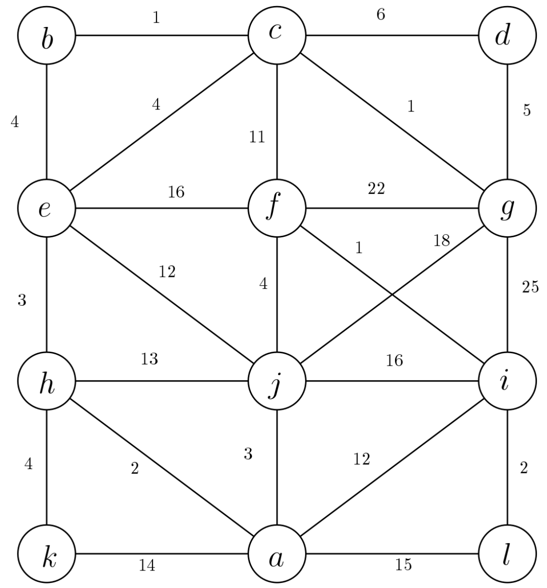
A. 5
B. $-\infty$
C. 3
D. 4
E. 2

31. The running time for the following pseudocode fragment is $f(n)$. Which of the following statements is correct?

function BLAH(positive integer n)
  $i \leftarrow 1$
  $j \leftarrow 1$
  while $i \leq n$ **do**
    while $j \leq n$ **do**
      Do constant number C of elementary operations
      $j \leftarrow 5j$
    $j \leftarrow 1$
  return something

A. $f(n)$ is $\Theta(n)$
B. $f(n)$ is $O(n)$
C. $f(n)$ is $\Theta(n \log_5(n))$
D. $f(n)$ is $\Theta(n^2)$
E. $f(n)$ is $\Omega(n^2)$

32. For the following edge-weighted graph, what is the minimum cost of a spanning tree?

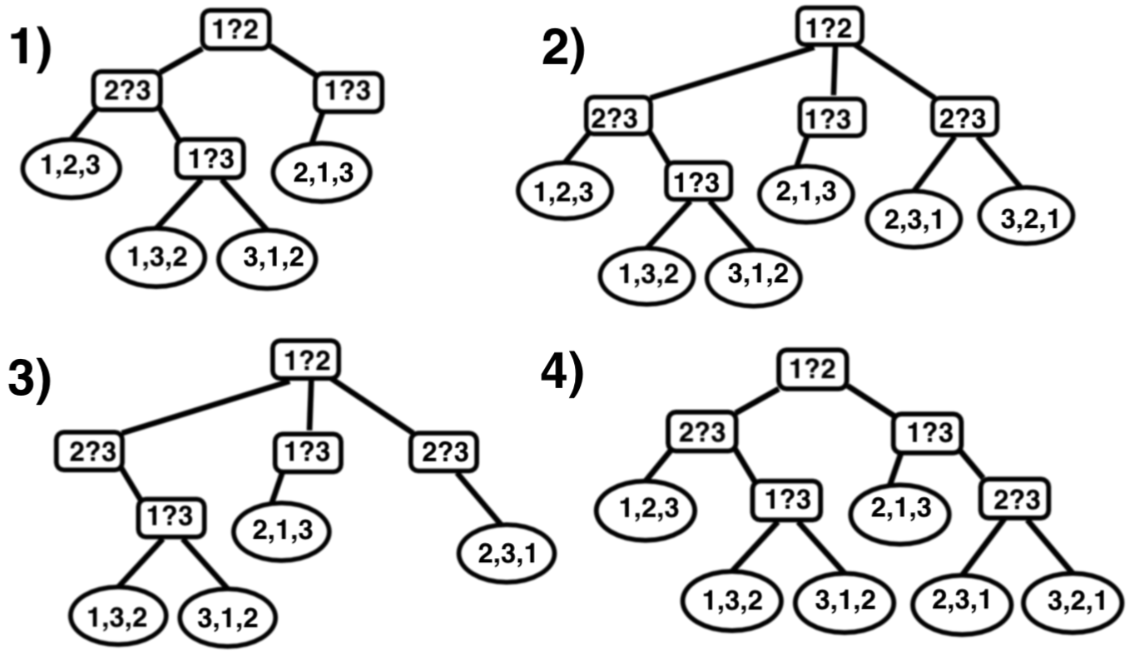


A. 34
B. None of the other answers.
C. 23
D. 33
E. 24

33. Which of the following examples demonstrates that Mergesort is not stable

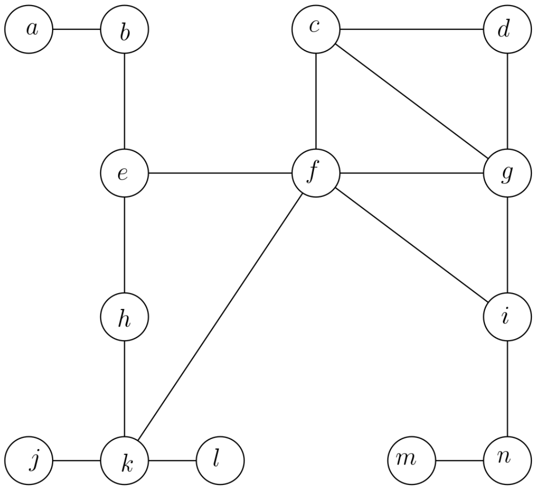
A. $[3, 4, 1_a, 1_b, 8]$
B. $[1_a, 1_b, 8, 3, 4, 1_c]$
C. None of the example because Mergesort is stable.
D. $[9, 2, 6, 1_a, 5, 6, 7, 1_b]$
E. $[1_a, 3, 7, 1_b, 9]$

34. Consider the four trees below. Which of these can be a decision tree of some comparison-based sorting algorithm on 3 elements?



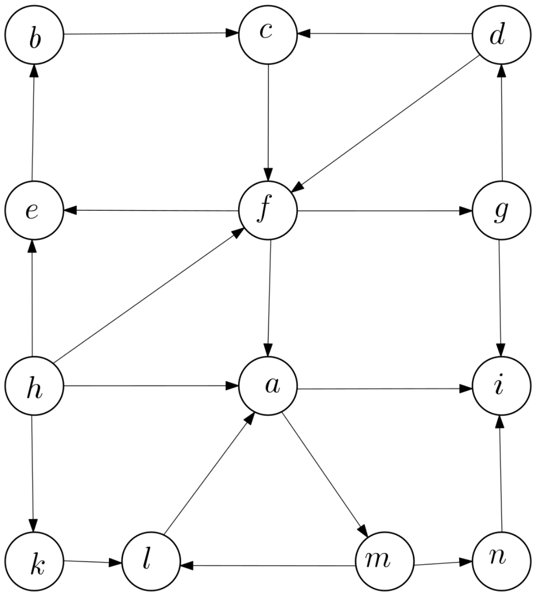
A. 2)
B. 1)
C. 4)
D. 3)
E. 2) and 4)

35. Consider the following graph. Do a BFS starting at vertex g. What is the 8th vertex visited (colored GREY). Assume that if there is a choice you choose the vertex with lower alphabetical order.



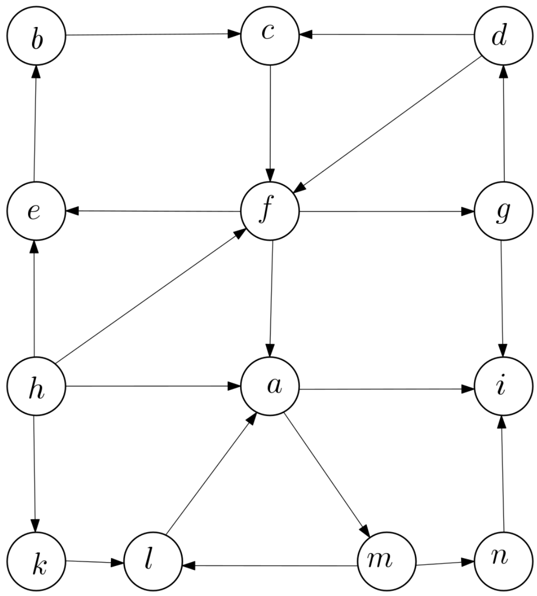
A. b
B. k
C. n
D. e
E. h

36. Consider the following digraph. What is its size?



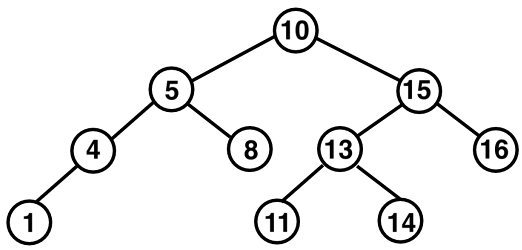
A. 22
B. 21
C. 13
D. 20
E. 14

37. Consider the following digraph. What is the length of the longest path?



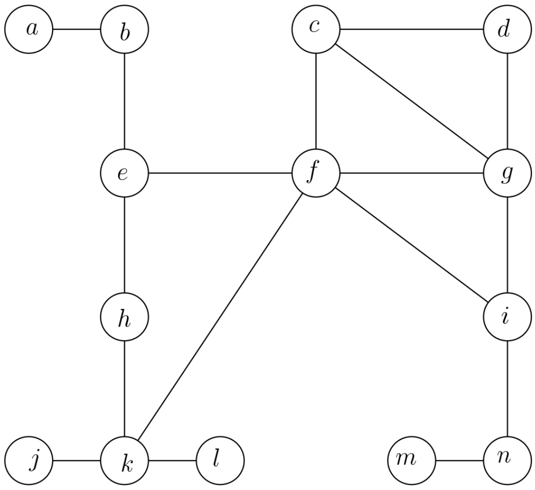
A. 5
B. 8
C. 4
D. 7
E. 6

38. Consider the binary search tree below. Which statement describes the process of inserting node 12 in the given BST?



A. Stop at the leaf 11 while searching for 12. Now draw 12 as the left child of 11.
B. Stop at the leaf 14 while searching for 12. Now draw 12 as the left child of 14.
C. Stop at the leaf 8 while searching for 12. Now draw 12 as the right child of 8.
D. Stop at the leaf 16 while searching for 12. Now draw 12 as the left child of 16.
E. Stop at the leaf 11 while searching for 12. Now draw 12 as the right child of 11.

39. Consider the following graph. What is its maximum degree?



A. 4
B. 5
C. 3
D. 2
E. 6

40. Which of the following statements describes the quicksort algorithm?

A. Split the input list into sorted and unsorted sublists. Repeat the following steps until the unsorted sublist is empty: find a maximal element of the unsorted part by sequential scan; move the maximal element to the head of the sorted part.
B. Choose a pivot p. Partition the list: the left sublist with elements $\leq p$ including the pivot, the right sublist with elements $\geq p$. Recursively sort left and right sublists.
C. Build the binary heap by heapifying the array, remove the highest priority node until heap is empty.
D. Split the input list into sorted and unsorted sublists. Repeat the following steps until the unsorted sublist is empty: choose the first element x in the unsorted sublist; find the last element $y \leq x$ in the sorted sublist; insert x after y in the sorted sublist.
E. Choose a pivot p. Partition the list: the left sublist with elements $\leq p$, the pivot, the right sublist with elements $\geq p$. Recursively sort left and right sublists.

Working page 1