COMP2521 Revision Trivia 3

Question

answer

Consider the following text and pattern:

Text:

A A A B C A A B A A B A A B A C A A B A

Pattern:

A A B A A B A C A

Search for the pattern in the text using the Knuth-Morris-Pratt algorithm. How many comparisons are made in total?

Question

answer

Suppose there was an empty hash table of size *N* = 7, which uses the hash function h(*x*) = *x* % 7. Insert these elements into the hash table in the given order, using linear probing to resolve collisions:

15, 10, 8, 5, 18, 3, 23

Question

answer

a) What is the best case time complexity for searching for an item in a hash table that has *N* items?

b) What would be the worst case time complexity?

c) How about the average case time complexity?

Question

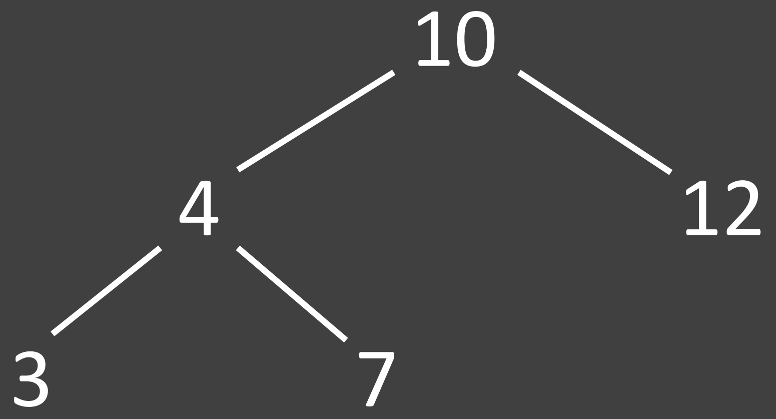
answer

Suppose you are given two sorting programs. You are told that one of them is selection sort and the other is insertion sort, but you are not told which one is which. Describe a test you could run to distinguish them.

Question

answer

Show the result of inserting an 8 in the following AVL tree.



Question

answer

Suppose we used an ordered singly linked list to implement a priority queue. What would be the time complexity of the enqueue and dequeue operations?

How does this compare to using a heap?

Explain your answers.

Question

answer

Suppose we had an empty hash table of size 11 that uses separate chaining, and the hash function h(*x*) = *x* % 11. Insert these elements into the table:

9, 17, 16, 5, 25, 20, 18, 11, 10, 3, 14

Which index(es) contain the longest chain, and what is their length?

Question

answer

Marc learned in COMP2521 that insertion sort and quick sort have an average time complexity of $O(n^2)$ and $O(n \log n)$ respectively. However, when he used them to sort random arrays of size 10, he found that insertion sort was consistently faster than quick sort. Does this contradict what he has been taught? Why or why not?

Question

answer

Construct the failure function for this pattern:

A E D A E A E D A A

Question

answer

Consider the following text and pattern:

Text:

E C B A A E D D A E D B A A E D C E A

Pattern:

E D B A A E D

Search for the pattern in the text using the Boyer-Moore algorithm. How many comparisons are made in total?

Question

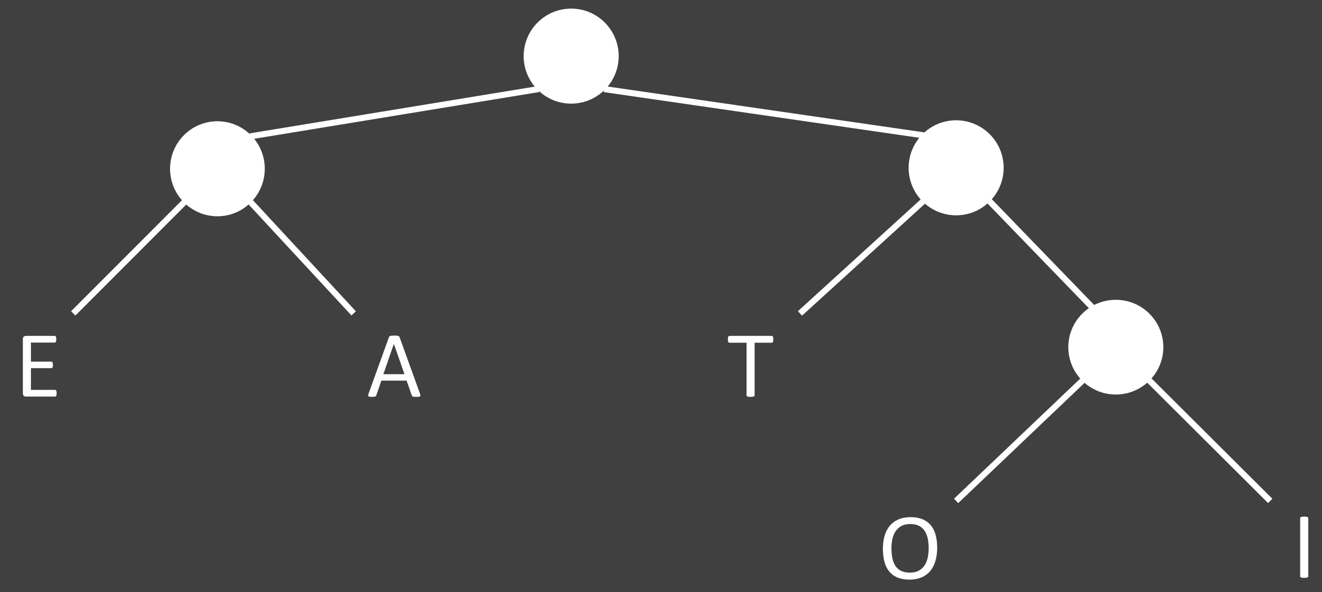
answer

Generally, more efficient algorithms tend to use more space than less efficient algorithms. This is known as the space-time tradeoff. With reference to two of the algorithms discussed in the course, give an example of this.

Question

answer

Suppose that after running the Huffman coding algorithm, you obtained this Huffman tree:



Show the Huffman code table.

Question

answer

Show how the following numbers after each iteration of sorting with an LSD (least-significant digit) radix sort, with a radix of 10.

4123, 5123, 4321, 4132, 1999

Question

answer

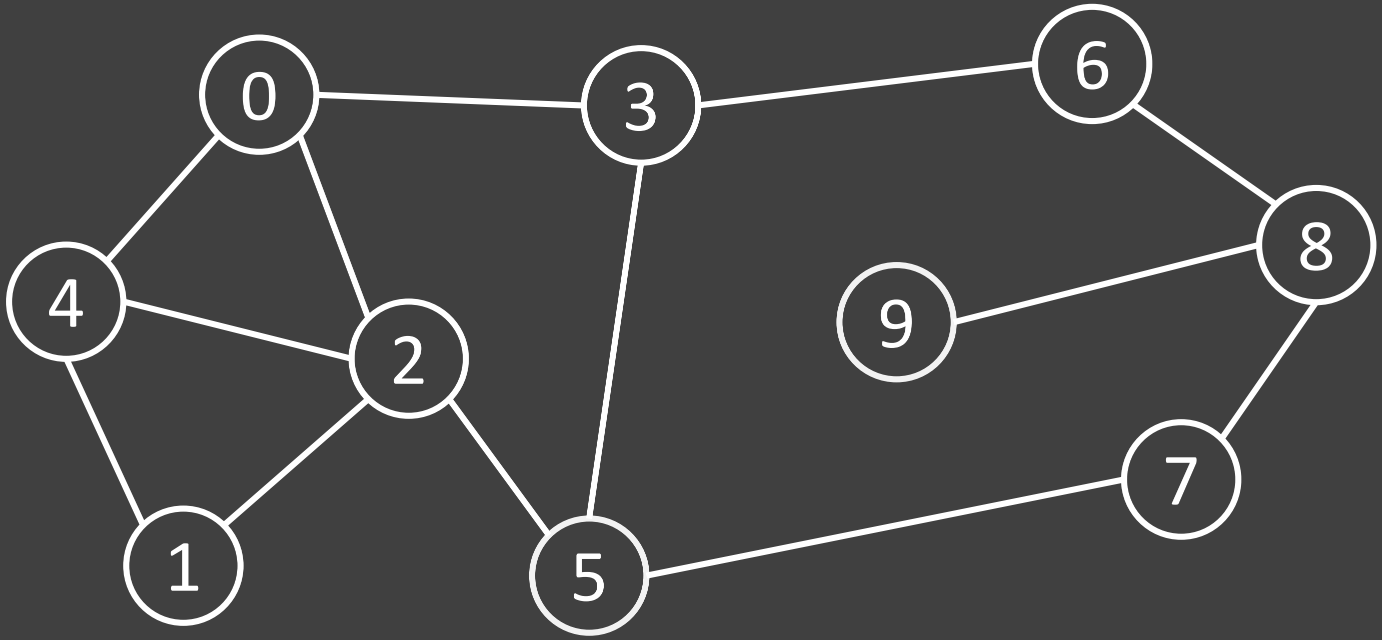
a) What is the minimum height of a binary tree with 8 nodes? Draw a possible 8-node tree with this height.

b) What is the maximum height of a binary tree with 8 nodes? Draw a possible 8-node tree with this height.

Question

Perform a DFS on this graph starting at vertex 0, and list the vertices as they are visited. If a vertex has multiple neighbours, visit the neighbour with the smaller vertex number first.

answer



Answers

Answer

Failure function:

A	A	B	A	A	B	A	C	A
0	1	0	1	2	3	4	0	1

```
A A A B C A A B A A B A A B A C A A B A
A A B A A B A C A
  A A B A A B A C A
        A A B A A B A C A
          A A B A A B A C A
                A A B A A B A C A
```

The pattern was found.
In total, 20 comparisons were made.

Answer

question

0	1	2	3	4	5	6
23	15	8	10	18	5	3

15, 10, 8, 5, 18, 3, 23

2

Answer

a) O(1)

b) O(*N*)

c) O(1)

Answer

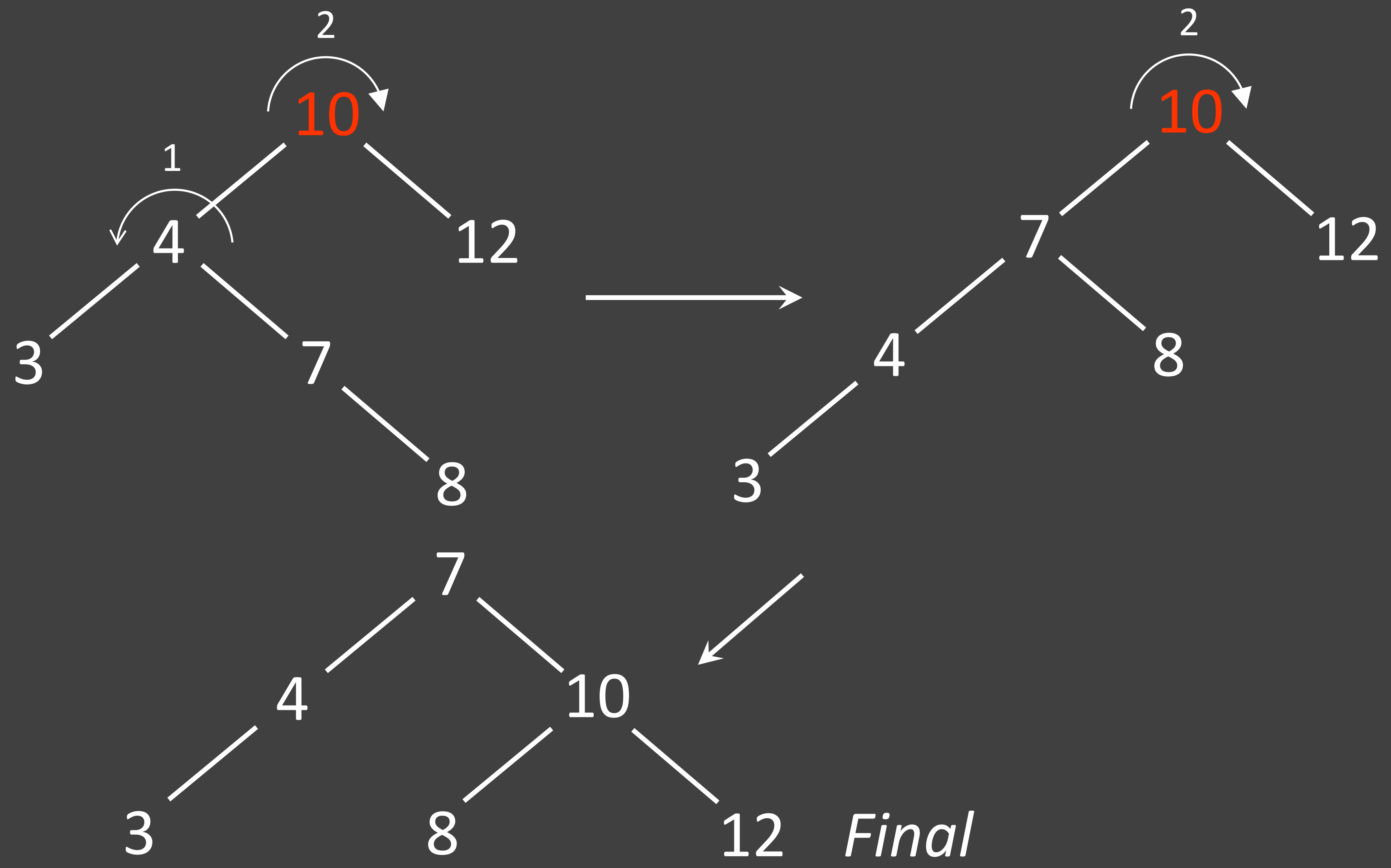
question

Two ways:

1) Insertion sort is stable, whereas selection sort is unstable. We can test the programs for stability by including different items with the same key in the input.
2) Insertion sort is adaptive, whereas selection sort is not. We can test the programs for adaptability by giving them sorted input of different sizes. Insertion sort will have a time complexity of O(N), while selection sort will have a time complexity of O(N^2) for already-sorted input.

Answer

question



Answer

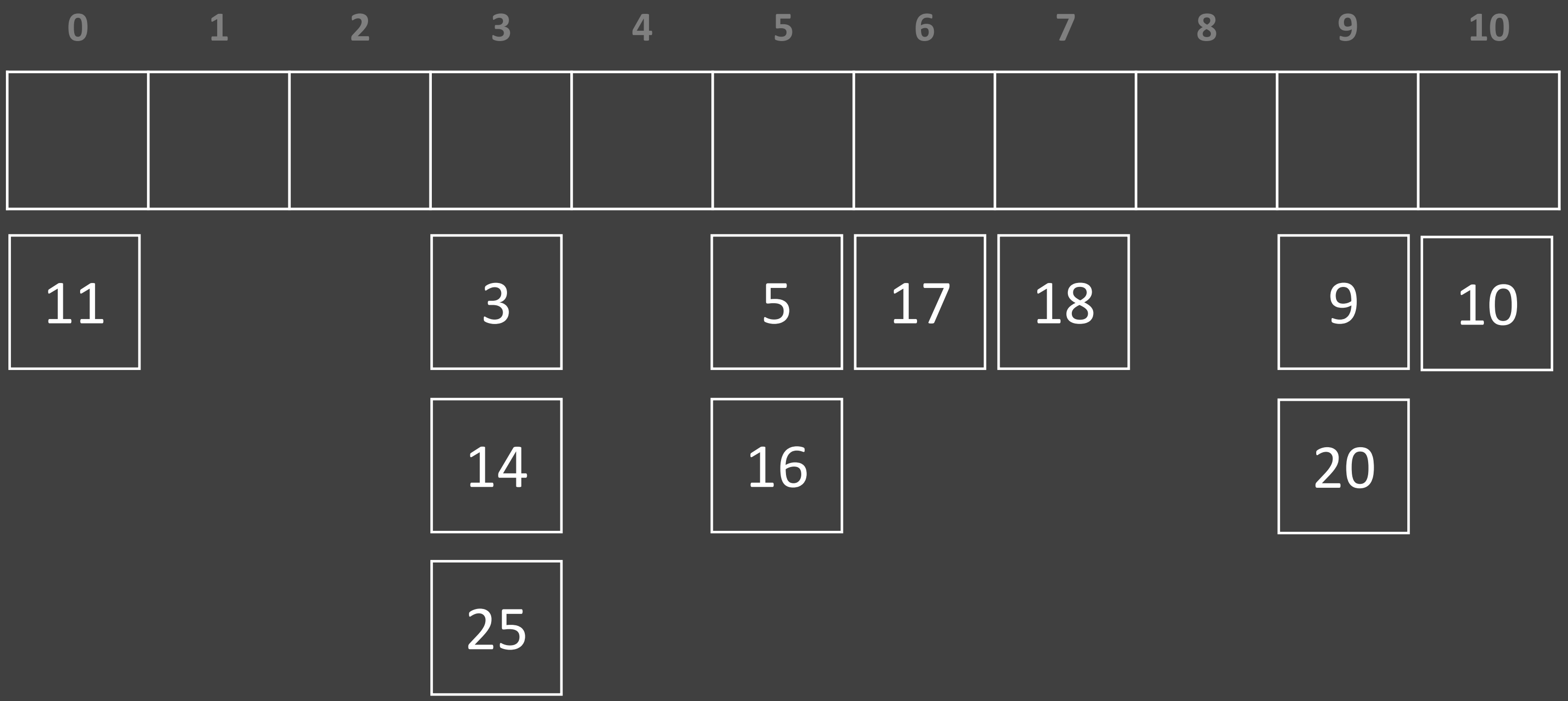
question

Enqueueing with an ordered linked list would be O(*N*), as you need to find the right place to insert the new item. Dequeuing would be O(1) as you can simply remove the first item in the list.

Enqueueing and dequeuing with a heap are both O(log *N*), as the values are arranged in such a way that they can be reasoned about as if they existed in a complete binary tree, and these operations involve traversing the height of the tree.

Answer

question



Index 3 has the longest chain. The length of the chain is 3.

9, 17, 16, 5, 25, 20, 18, 11, 10, 3, 14

Answer

question

No. Time complexity describes the growth rate of the time taken as the input size increases, not how long an algorithm or program will take to run. So even though insertion sort has a ‘worse’ time complexity than quick sort, this does not mean that it will take longer than quick sort for all inputs. However, it does guarantee that eventually (i.e., when the input is sufficiently large), insertion sort will take longer.

Answer

question

Failure function:

A	E	D	A	E	A	E	D	A	A
0	0	0	1	2	1	2	3	4	1

A E D A E A E D A A

Answer

Last occurrence function:

x	A	B	C	D	E
L(*x*)	4	2	-1	6	5

question

```
E C B A A E D D A E D B A A E D C E A
E D B A A E D               L(C) = -1
    E D B A A E D           L(A) =  4
        E D B A A E D       L(D) =  6
          E D B A A E D     L(B) =  2
                  E D B A A E D
```

The pattern was found.

In total, 19 comparisons were made.

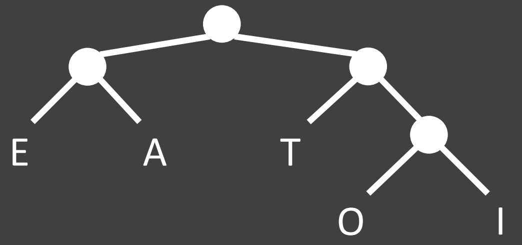
Answer

question

An example is merge sort and insertion sort. Merge sort has a worst case time complexity of O(n log n) and is more efficient than insertion sort, which is O(n^2) in the worst case. However, merge sort has a space complexity of O(n), whereas insertion sort uses constant (O(1)) space.

Answer

A	01
E	00
I	111
O	110
T	10



Answer

question

4123, 5123, 4321, 4132, 1999

4321, 4132, 4123, 5123, 1999

4321, 4123, 5123, 4132, 1999

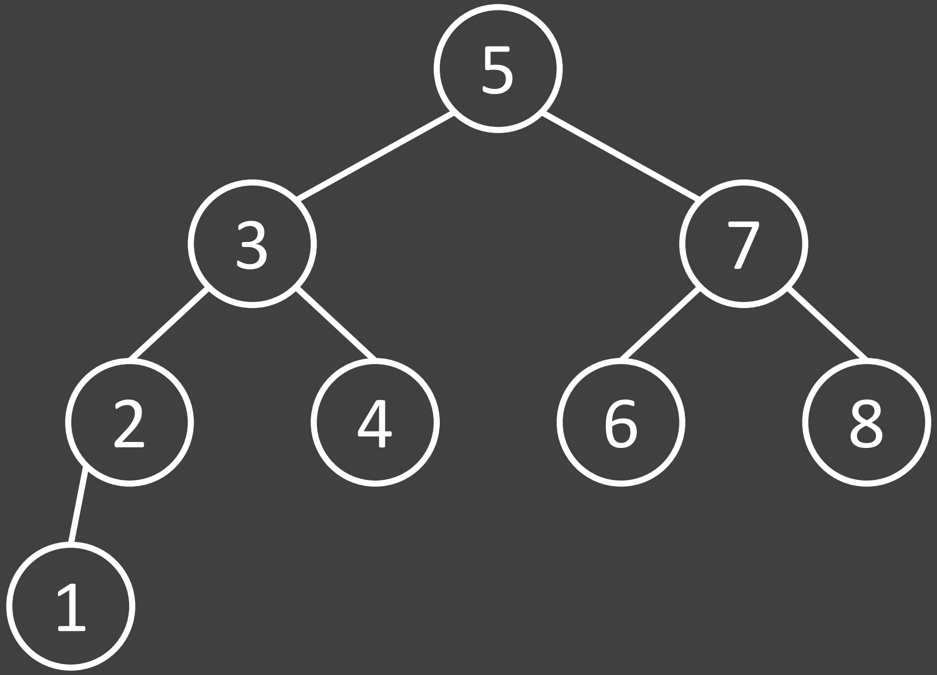
4123, 5123, 4132, 4321, 1999

1999, 4123, 4132, 4321, 5123

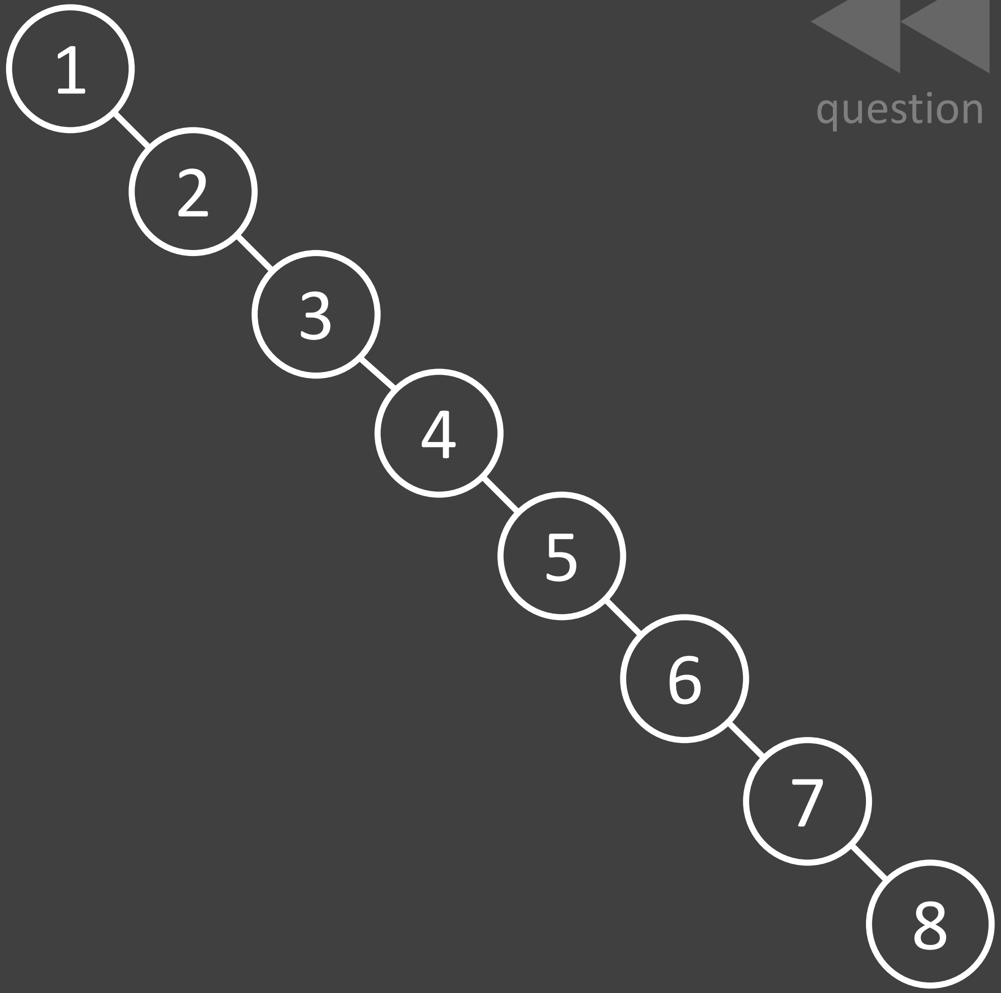
Answer

question

a) 3



b) 7

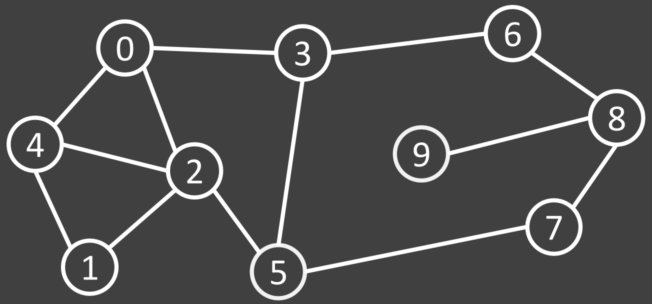


Answer

question

DFS starting at vertex 0:

0 2 1 4 5 3 6 8 7 9


0
3
6
4
2
9
8
1
5
7