Q5 (through Graph Search 2)

This is a preview of the published version of the quiz

Started: Nov 2 at 9:52am

Quiz Instructio

Question 1 — 0.1 pts

Which acronym does NOT refer to a type of graph?

- DAG
- BST
- BFS

Question 2 — 0.1 pts

What is the largest number of grandchildren that the root of a binary tree might have?

Question 3 — 0.1 pts

Which statement is TRUE?

- every binary tree is a BST

- every DAG is also a directed graph
- every non-root in a binary tree has exactly two parents
- every non-leaf in a binary tree has exactly 2 children

Question 4 — 0.1 pts

In order to make worst-case lookups as fast as possible, we prefer our BSTs to be:

- short
- tall

Question 5 — 0.2 pts

During a graph search, the program crashes due to a stack overflow (that is, too many stack frames are allocated). What graph search algorithm is being used?

- DFS
- BFS

Question 6 — 0.1 pts

For which use case is Python's deque data structure NOT suitable?

- queue

- stack
- priority queue

Question 7 0.1 pts

Which of the follov always a DAG?

- streets: nodes for intersections, edges for street sections connecting intersections
- human ancestry: nodes for people, edges from parents to children
- git: nodes for commits, edges pointing from each commit to the prior commit

Question 8 0.2 pts

Assume nums is a list with N elements. What is the complexity of the following code? Calling len(q) is a single step. For the complexity of the other deque operations in the following code, it may be useful to reference the first couple paragraphs of the deque documentation: https://docs.python.org/3/library/collections.html#collections.deque (https://docs.python.org/3/library/collections.html#collections.deque) before answering.

If multiple answers are correct, choose the best answer.

```
from collections import deque

q = deque()

for x in nums:
    q.append(x)

while len(q) > 0:
    print(q.popleft())
```

- O(1)

- ○ O(N)
- ○ O(N + (N-1) + (N-2) + ... + 2 + 1)
- ○ O(N**3)
- ○ O(N**2)
- ○ O(N + N**2)

Not saved Submit Quiz