

False, a red black tree must have a black root node. If you pick the left subtree of the root node, then the “new” root will be red. This does not meet the criteria for a red black tree.

True, external nodes are black, so the child of an external node must be red.

$$\log_2(n) \rightarrow 16.6096 \dots \rightarrow 16$$
$$\log_2(n) \rightarrow 16.6096 \dots \rightarrow 16$$

$O(n) \rightarrow 100000$

$O(n) \rightarrow 100000$

-inf	-----										+ inf
-inf	----- 17 -----										+ inf
-inf	----- 17 ----- 42 -----										+ inf
-inf	-- -- -- 17 -- -- -- 24 -- -- -- 42 -- -- --										+inf
-inf	----- 17 ----- 24 ----- 31 ----- 42 -----										+ inf
-inf	-12 -- 17 -- -- 24 -- 31 ----- 42 -----										+ inf
-inf	- 12 -- 17 -20 -- 24 -- 31 -- 39 ----- 42--44--48 - 50										+ inf

4

[illegible]

5

0	1	2	3	4	5	6	7	8	9	10
13	94	39	16	5	44	88	11	12	23	10

6

0	1	2	3	4	5	6	7	8	9	10
		44		20	16	12	13			
		88			5	23				
		11				94				
						39				

7 $\log(n) * \log(n)$

8

A Queue would be the most beneficial data structure to meet the project criteria.
Assuming the timestamp is always increasing, the order in which flights are processed is based on a First In First Out implementation.

9

13.3

H <- A <-> B <-> C <-> D <-> E <-> F <-> G <-> H -> A

traversal: A,B,C,D,E,F,G,H ... from H go the other way "without lift pencil" ... H,G,F,E,D,C,B,A

13.4

A <-> B <-> C <-> D <-> E <-> F <-> G <-> H -> A

traversal: H,A,B,C,D,E,F,G,H ... from H go the other way "without lift pencil" ... H,G,F,E,D,C,B,A

10

- a. Adjacency list — saves space
- b. Adjacency list — saves space
- c. Adjacency Matrix — easy & quick visualization and access

11

Root of tree is node that begins Dijkstra's algorithm iterations. Then based on weights displayed in final iteration of the table, construct a tree in order of least weight to greatest weight

13 (Edges are when student gets a C or greater grade from parent)

[University Core curriculum] — CSCE 121

/ \
CSCE 222 CSCE 221
 \ /

Upper Level (300's)

14 One of each vertex will be printed, since after discovery, that node isn't traversed again.

- 15
- a read/write: $\log(n)$
 - b read: $O(n)$, write: $O(1)$
 - c read: $O(1)$, write: $O(n)$