Multiple Choice: choose the best answer for each.

(2 pts) 1. Which version of UF (Union-Find) has the best worst-case time for the find operation?
   A. quick-find  B. quick-union  C. weighted-union  D. path-compression

(2 pts) 2. Which version of UF has the best worst-case time for the union operation?
   A. quick-find  B. quick-union  C. weighted-union  D. path-compression

(2 pts) 3. Who invented our method for deletion in a BST (binary search tree)?
   A. Floyd  B. Hibbard  C. Knuth  D. Sedgewick  E. Tarjan

(2 pts) 4. In the particle system simulation, what data structure stores the future events?
   A. array  B. BST  C. hash table  D. heap  E. UF

(2 pts) 5. Which is the best estimate for $N \lg N - \lg(N!)$ ?
   A. $O(N \lg N)$  B. $O(N)$  C. $O(\sqrt{N})$  D. $O(\lg N)$  E. $O(1)$

(2 pts) 6. What is the time for TopM to find the $M$ largest among $N$ inputs?
   A. $O(M \lg M)$  B. $O(M \lg N)$  C. $O(N)$  D. $O(N \lg M)$  E. $O(N \lg N)$

(2 pts) 7. Where does BFS keep track of the vertices it needs to visit next?
   A. explicit stack  B. runtime stack  C. fifo queue  D. priority queue  E. marked array

(2 pts) 8. What edge type cannot occur in the DFS traversal of an undirected graph?
   A. back  B. cross  C. forward  D. tree

(2 pts) 9. What edge type cannot occur in the DFS traversal of a DAG (directed acyclic graph)?
   A. back  B. cross  C. forward  D. tree

(2 pts) 10. Which is NOT found using DFS?
    A. bridges  B. connected components  C. cycles  D. shortest paths  E. topological order

(2 pts) 11. On an input graph (or digraph) with $V$ vertices and $E$ edges, which algorithms use $O(V)$ extra space? (That is, memory beyond that for the input.)
   Choose TWO:  A. BFS  B. Kosaraju-Sharir  C. Kruskal  D. lazy Prim  E. eager Prim

(2 pts) 12. Which red edge arrangements cannot occur during insertion in a left-leaning red-black tree?
   Choose TWO:  A.  B.  C.  D.  E.
Fill in the Blank: partial credit is sometimes possible.

(2 pts) 13. Suppose \( uf \) is a UF data structure, providing only the methods \( uf\text{.}find(p) \) and \( uf\text{.}union(p,q) \). What boolean Java expression tests whether \( p \) and \( q \) are already connected?

13. \( uf\text{.}find(p)==uf\text{.}find(q) \)

(2 pts) 14. Both binary search trees and hash tables are examples of this more general kind of data structure (mapping keys to values).

14. symbol table, ST

(2 pts) 15. What kind of persistence is it, if old versions of a data structure are searchable, but you can only modify the most recent version?

15. partial

(2 pts) 16. The BST \( \text{rank} \) and \( \text{select} \) methods rely on what extra data field (beyond key, value, left, right) being present in each Node? (We saw it in PBST.)

16. \( N, \text{size} \)

(2 pts) 17. Java collection iterators (for example, the \texttt{java.util.TreeMap} iterator) can “fail-fast”, by checking that what field still has its expected value?

17. \( \text{modCount} \)

(2 pts) 18. Which kind of hashing allows the load factor \( \alpha \) to exceed one?

18. \( \text{chaining} \)

(2 pts) 19. We observe “clustering” in what kind of hashing?

19. \( \text{linear probing} \)

(2 pts) 20. Kruskal’s algorithm and Prim’s algorithm are both special cases of what general MST algorithm?

20. \text{greedy}

(2 pts) 21. In the book code, both Dijkstra’s algorithm and the eager Prim algorithm use this variant of the binary min heap.

21. \text{IndexMinPQ}

(2 pts) 22. In both Dijkstra’s algorithm and the eager Prim algorithm, if we replace the binary heap by a “Fibonacci heap”, then what big-Oh bound do we get on the total running time?

22. \( O(V\lg V+E) \)

(2 pts) 23. A graph is bipartite if and only if it does not contain what?

23. \( \text{odd cycle} \)

(2 pts) 24. Suppose we have a data structure \( S \) and potential function \( \phi(S) \). For an operation \( o \), we define its amortized time \( T_{\text{amortized}}(o) \) to be its actual time \( T_{\text{actual}}(o) \), plus what?

24. \( \phi(S_{\text{after}}) - \phi(S_{\text{before}}) \) (variants OK)
Short Answer.

(6 pts) 25. Consider the LLRB (left-leaning red-black) tree below, thicker edges are red. First, draw the corresponding 2-3 tree. Second, draw the LLRB tree that results from inserting E.

![LLRB tree and corresponding 2-3 tree](image)

(6 pts) 26. Suppose we do a full DFS traversal of the DAG below (so we start at 0, and make sure each each vertex is eventually marked). Suppose each adjacency list is sorted (so, the first edge out of 1 is to 2). First, mark the “tree” edges. Second, list the vertex ids in the topological order found by this traversal.

![DAG for DFS traversal](image)

Topological Order (= reverse postorder):
3, 7, 4, 6, 1, 5, 8, 9, 6, 2

(6 pts) 27. Consider the edge weighted graph below. List its MST edges in the order found by Kruskal’s algorithm, and in the order found by Prim’s algorithm. (Name edges by their weights, use 0 as a start vertex.)

![Edge weighted graph](image)

Kruskal: 1, 2, 3, 4, 6, 7, 8, 10, 15

Prim: 7, 10, 8, 4, 3, 6, 2, 1, 15

(6 pts) 28. Consider cuckoo hashing, and hashing by linear probing. List an advantage of cuckoo hashing, and an advantage of linear probing.

Cuckoo advantage: O(1) worst-case find (and delete)
Probing advantage: still works with a poor hash function (does not need a family)

Other answers may also suffice ("space" is not full credit, since they are similar)