This exam is no book, no gadgets. You are allowed one sheet of notes (letter size, front and back). You have the full period (50 minutes). There are 23 questions, worth 56 total points. Marks will be curved so the median grade is at least a B. Note: we are not doing the "sheet of notes" for our Spring 2016 midterm.

Name (Print):

__________________________________________________________

Signature:

This exam is my own work. I understand it is governed by the Emory Honor Code.

Multiple Choice: choose the best answer for each.

(2 pts) 1. Which version of Union-Find requires the most space?
   A. quick-find  B. quick-union  C. weighted quick-union  D. path compression
   E. half path compression

(2 pts) 2. Which algorithm did we use to solve the Kevin Bacon game?
   A. BFS  B. DFS  C. Dijkstra  D. Heapsort  E. Prim

(2 pts) 3. What is found by the Kosaraju-Sharir algorithm?
   A. bridges  B. components  C. cycles  D. SCC's  E. shortest paths

(2 pts) 4. Left-leaning red-black trees can be thought of as representing:
   A. 2-3 trees  B. 2-4 trees  C. complete trees  D. random trees  E. union-find trees

(2 pts) 5. Who worked out the expected time analysis for linear probing?
   A. Floyd  B. Hibbard  C. Knuth  D. Sedgewick  E. Tarjan

(2 pts) 6. Where does DFS keep track of the vertices it needs to finish visiting?
   A. explicit stack  B. runtime stack  C. fifo queue  D. marked array  E. priority queue

(2 pts) 7. What is an advantage of a $d$-ary min-heap over a binary min-heap?
   A. faster sink  B. faster swim  C. faster insert  D. less space  E. modifiable keys

(2 pts) 8. What is an advantage of IndexMinPQ over a binary min-heap?
   A. faster sink  B. faster swim  C. faster insert  D. less space  E. modifiable keys

(2 pts) 9. What is an advantage of linear probing over chaining?
   A. allow larger $\alpha$  B. easier analysis  C. easier deletion  D. fewer collisions  E. less space

(2 pts) 10. What time bound did we argue for the quickselect algorithm on $N$ items?
    A. $O(N)$ expected  B. $O(N)$ worst-case  C. $O(N \log N)$ expected  D. $O(N \log N)$ worst-case
    E. $O(N^2)$ expected
Fill in the Blank: partial credit is sometimes possible.

(2 pts) 11. What is our lower bound on comparisons required to sort \( N \) items? (Preferably without \( O/\Omega/\Theta \).)

11. \( \lg(N) \)

(2 pts) 12. Suppose we store a binary heap in an array, with the root node at index 0. If a node is at index \( j \), the left child of that node is at what index?

12. \( 2j+1 \)

(2 pts) 13. Given \( K \) sorted lists, containing \( N \) items overall, multiway merge takes how much time? (Use big-Oh.)

13. \( O(N \lg K) \)

(2 pts) 14. Which hashing method can search in worst-case \( O(1) \) time?

14. cuckoo hashing (or perfect)

(2 pts) 15. Which hashing method allows load factor \( \alpha \) larger than one?

15. chaining

(2 pts) 16. What is our name for the order in which vertices are first discovered by a DFS traversal?

16. preorder

(2 pts) 17. In hw2, what is the worst-case time for a StackIterator to visit all \( N \) keys in its tree? (Use big-Oh.)

17. \( O(N) \)

(2 pts) 18. Given a graph with \( V \) vertices and \( E \) weighted edges, how much time is used by Kruskal’s algorithm? (Use big-Oh.)

18. \( O(E \lg V), \text{ or } O(E \lg E) \)

(2 pts) 19. Given a graph with \( V \) vertices and \( E \) weighted edges, how much extra space (beyond the graph itself) is used by the eager Prim algorithm? (Use big-Oh.)

19. \( O(V) \)

(2 pts) 20. In hw3, after the first DFS pass, suppose we have a tree edge from \( v \) to \( w \). What boolean expression tells us whether this is a bridge?

20. \( \text{pre}[w] = \text{low}[w] \)
Short Answer.

21. (6 pts) Consider the edge weighted graph on the board. List the order that its MST edges would be found by Kruskal’s algorithm, and the order that its MST edges would be found by Prim’s algorithm. (You can name the edges by their weights.)

- Depends on the graph.
- Similar question: what edges are found in the first round of Boruvka?

(6 pts) 22. Compare our persistent red-black tree with the book’s (mutable) red-black tree. Give at least one advantage (preferably two) of each over the other.

- Ours: persistence (access to old versions), iterator cannot fail, no need for locking
- Book: less total space (only store most recent tree), faster by constant factor

(4 pts) 23. Suppose we have a graph $G$ with weighted edges, and $e$ is a lightest edge touching some vertex $v$. Suppose $T$ is a spanning tree of $G$ that does not contain $e$. Argue there is another spanning tree $T'$ that contains $e$, and is no heavier than $T$.

- Add $e$ to $T$, it forms a cycle, and that cycle has some other edge $f$ also adjacent to $v$. Now $T' = T + e - f$ is another spanning tree, and it is no heavier than $T$ since $w(f) \geq w(e)$. 