This exam is no book, no gadgets. You are allowed to use one sheet of notes (letter size, two sides), which you should turn in with the exam. There are 32 questions, worth 67 total points. Marks will be curved so the median grade is at least B.

Name (Print): 

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

Signature: 

Choose an implementation: choose the most appropriate algorithm for each situation.

(2 pts) 1. Finding a shortest route to a destination, using a road map with road lengths.
   A. BFS  B. Dijkstra  C. Acyclic  D. Bellman-Ford

(2 pts) 2. Seam carving: removing paths of pixels, to shrink an image.
   A. BFS  B. Dijkstra  C. Acyclic  D. Bellman-Ford

(2 pts) 3. Finding a shortest augmenting path in the residual graph, for Ford-Fulkerson.
   A. BFS  B. Dijkstra  C. Acyclic  D. Bellman-Ford

(2 pts) 4. Scheduling jobs with precedence constraints (critical path method).
   A. BFS  B. Dijkstra  C. Acyclic  D. Bellman-Ford

(2 pts) 5. Finding a profitable trading cycle in a currency market.
   A. BFS  B. Dijkstra  C. Acyclic  D. Bellman-Ford

(2 pts) 6. For sorting general text strings of varying length, and perhaps big $R$ (according to Sedgewick).
   A. key-indexed counting  B. LSD  C. MSD  D. 3-way quicksort  E. heapsort

(2 pts) 7. For sorting a hundred million 32-bit integers, thinking of each as a string of length two ($R = 2^{16}$).
   A. key-indexed counting  B. LSD  C. MSD  D. 3-way quicksort  E. heapsort

(2 pts) 8. For sorting many long random bitstrings, of varying length ($R = 2$).
   A. key-indexed counting  B. LSD  C. MSD  D. 3-way quicksort  E. heapsort

Fill in the Blank: fill each blank appropriately.

(2 pts) 9. In hw4, what trick did we propose for building the QuadTree, to make its expected depth $O(\log N)$?
   
   9. random insert order

(2 pts) 10. What is the main advantage of the TST over the R-way trie?
   
   10. space

(2 pts) 11. What is a potential (rare) disadvantage of the TST, compared to the R-way trie?
   
   11. unbalanced (slow)

(2 pts) 12. A “Patricia” trie is like a standard trie, but improved in what way?
   
   12. compressed
(2 pts) 13. Give an example of a string-based operation that can be handled efficiently using a trie, but not by a generic symbol table (such as a red-black tree).

13. \text{longestPrefixOf} (s) \left[ \begin{array}{c} p_a g r e e \end{array} \right]

(2 pts) 14. Given \( N \) points in the plane, their Delaunay triangulation contains an edge \( pq \) (that is, the line segment between points \( p \) and \( q \)), whenever what condition is true?

\text{Skep!}

(2 pts) 15. In hw4, our first tour was a preorder traversal of the EMST. Its length has what upper bound, in terms of \( \text{OPT} \) (the optimal tour length)?

15. \text{OPT}

(2 pts) 16. Given a string of length \( N \), Manber's algorithm can sort all its suffixes (or cyclic shifts). What is its \( \text{big-Oh} \) worst-case running time bound?

\text{expected over all inputs} \text{ worst-case overall input}

16. \( O(N \log N) \)

(2 pts) 17. Give an example of a string-based operation that can be handled efficiently using a trie, but not with a generic symbol table (such as a red-black tree).

\text{RESPECT!}

(2 pts) 18. What is the main advantage of the KMP NFA over the KMP DFA?

18. \( O(\text{MR}) \) less space/time to build \( O(M) \)

(2 pts) 19. Suppose we do Huffman compression of the text "PUTATOPUT" (without the quotes). What is the length, in bits, of the compressed text?

19. 18

(2 pts) 20. Continuing the previous question, what is the length of the bitstring that describes the code trie? Assume 8 bits per character.

20. 39

(2 pts) 21. Suppose we do LZW compression, with our usual initial code setup: \( 41 \) is \( A \), \( 42 \) is \( B \), \( 80 \) is end-of-string, and \( 81 \) is the first available code. What sequence of codes is output, when compressing "ABBABBA"?

21. \( 41, 42, 42, 81, 81, 80 \)

(2 pts) 22. Suppose we do LZW expansion, with the same initial setup. What string do we recover from the code sequence \( 42, 41, 81, 83, 81, 80 \)?

22. \( B A B A B A B A \)

(2 pts) 23. What is the BWT transform of "papaya", adding the mark character '%'? Note that '%' precedes the other characters, in sorted order.

23. \text{a}\text{ypp}\text{pa}\text{ap}\text{ya}\text{a}\text{p}\text{a}\text{a}\text{p}\text{a}\text{a}\text{p}\text{a}\text{a}\text{p}\text{a}
(2 pts) 24. What is the reverse BWT transform of "arrfowl", removing the mark character 'k'? (Hint: his name is on two of our algorithms.)

(2 pts) 25. Suppose we have a flow network $G$, with a max flow $f$ and a min cut $(A, B)$ (so $s \in A$, $t \in B$). What can we say about $f(e)$, for an edge $e$ that goes from $A$ to $B$?

(2 pts) 26. Suppose we use the "shortest path" rule to pick augmenting paths in Ford-Fulkerson. What upper bound do we have on the number of augmenting paths needed? (Suppose $V$ vertices and $E$ edges.)

(2 pts) 27. Suppose we apply Ford-Fulkerson, as described by the book, to the problem of matching $N$ students with $N$ jobs (bipartite matching). Suppose there are $M$ "compatibility" edges between students and jobs. What upper bound can you put on the number of augmenting paths needed? (You can avoid big-Oh here.)

(2 pts) 28. Given a flow network $G$ with source $s$ and sink $t$, we can ask for the maximum value of a flow from $s$ to $t$. With linear programming (LP), we can express this as a maximization problem. What problem is expressed by the dual LP?

(2 pts) 29. Give an example of a problem (from book, slides, or elsewhere) that can be reduced to linear programming. I prefer something that is not otherwise mentioned on the exam.

Short Answer:

(3 pts) 30. Draw the KMP DFA for the pattern ABAAC (assume the only possible input characters are A,B,C).

(3 pts) 31. Suppose we sort the following list of six words using MSD radix sort. Indicate (perhaps by underlining) which characters were not inspected.

(3 pts) 32. Suppose we insert those six words (in the order above) into a TST, draw the resulting structure. Do naïve insertions, no rebalancing, just keys (no values needed).