CS477/677 Analysis of Algorithms
Sample Final Exam

Name:____

1. **[30 points]** True/False Questions – To get credit, **you must give brief reasons for each answer!**

T F Given that a graph contains non-negative weights, there is no particular advantage using Dijkstra's algorithm vs Bellman-Ford's algorithm in solving the shortest paths problem.

T F Prim's algorithm is always asymptotically faster than Kruskal's algorithm.

T F Finding whether a graph contains a universal sink or not can be done in $O(V^2)$ time assuming adjacency matrix representation.

T F Given a weighted directed graph with distinct weights, the shortest path between any two vertices will be unique.

T **F** If A is in NP, then A is not in P.

T **F** A is an array of integers in non-increasing order. A represents a valid heap.

T **F** Counting-Sort sorts any sequence of inputs in O(n) time.

T **F** A red-black tree can be used to sort n inputs on O(nlgn) time.

T **F** A MST can contain negative cycles.

T **F** Retrieving an element using hashing with collisions resolved by chaining takes O(1) time on the average.

2. **[10 points]** Graphs can be represented using adjacency lists or adjacency matrices. What are the memory requirements of each representation? Discuss some advantages/disadvantages for each representation (i.e., situations where one representation is more suitable than the other).

3. [10 points] Prove the correctness of Dijkstra's algorithm. Can your proof go through without assuming non-negative weights? Justify your answer.

4. **[10 points]** Let G=(V,E) be any weighted connected graph. If C is any cycle of G, then show that the heaviest edge of C cannot belong to a MST of G.

5. **[10 points]** Briefly describe how radix sort works. Can we implement radix sort using any sorting algorithm? What is the running time of radix sort? How does the choice of the radix affect running time?

6. [10 points] This problem is on hashing.

(a) Demonstrate the insertion of keys 5, 28, 19, 15, 20, 33, 12, 17, and 10 into a hash table with collisions resolved by chaining. Let the table have 7 slots and the hash function be h(k)=k mod 7.

(b) Analyze the average performance of hashing with chaining (i.e., how long does it take to search for a key on the average using hashing with chaining?). State clearly the assumption being made in your analysis.

7. **[10 points]** Prove that a red-black tree with n internal nodes has height at most $2\lg(n + 1)$.

8. (Undergraduate Students only) [10 pts] Your friend Bob claims that he has "another" algorithm that can print out n keys stored in an n-node heap in sorted order, using only O(n) comparisons. Can you prove that he must be wrong? (*hint*: do not base your answer on the fact that printing out the keys takes O(nlgn) time according to the textbook's algorithm; remember, Bob's algorithm is different)

8. **(Graduate Students only) [10 pts]** Prove that any comparison sort algorithm requires $\Omega(n\lg n)$ comparisons in the worst case.