Math 320 Midterm 1 Study Guide

1. General information

- The exam will cover all sections from chapters 1–4 that were covered in class.
- The exam will be in the testing center from Wednesday, October 12 through Friday October 14. Check the testing center hours and lines, and be sure to give yourself enough time to finish.
- Books, notes, calculators, and computers will not be allowed.
- This study guide is not exhaustive. Not appearing on the study guide does not mean that something will not appear on the exam.

2. Definitions

Know the definitions discussed in the book and in class, including:

(1) Big-O and little-o notation
(2) Temporal and spatial complexity of an algorithm
(3) Asymptotic equivalence
(4) Difference operator Δ
(5) Cartesian product $A \times B$
(6) $P(n, r)$, $C(n, r)$
(7) $a|b$
(8) $\gcd(a, b)$
(9) $a \equiv b \pmod{n}$
(10) $\Gamma(x)$
(11) Directed and undirected graphs (as sets, pictures, or matrices)
(12) Walks, paths, cycles, connected graphs
(13) Trees, forests
(14) Linked list, stack, queue
(15) Binary search tree, AVL tree
(16) Priority queues, binary heaps
(17) Dynamic programming: top down and bottom up
(18) Bellman's optimality principle
(19) DFS, BFS
(20) Dijkstra's, Prim's, and Kruskal's algorithms
(21) Encoding scheme
(22) Instantaneous and uniquely decipherable encoding schemes
(23) Average word length in an encoding scheme
(24) Huffman encoding
(25) P, NP, NP-complete, NP-hard
(26) Traveling salesman and knapsack problems

Be able to produce examples and non-examples for these definitions.

3. Theorems you should know and be able to state and be able to use

Be sure to know the full statement of each theorem, including all hypotheses.

(1) Master Theorem
(2) Division Theorem

(3) Substitution rule for modular arithmetic
(4) Fundamental Theorem of Finite Calculus
(5) Binomial theorem
(6) Pascal's rule
(7) Fermat's little theorem
(8) Stirling's approximation

4. Algorithms you should know and be able to use

When appropriate, you should know the temporal and spatial complexity of these algorithms.

(1) Binary search
(2) Merge sort
(3) Fast multiplication
(4) Fast modular exponentiation
(5) Euclidean algorithm
(6) DFS, BFS
(7) Dijkstra's, Prim's, and Kruskal's algorithms
(8) Huffman encoding

5. Sample problems

(1) Given the input data, describe each step in one of the algorithms above.
(2) Prove that asymptotic equivalence is an equivalence relation.
(3) Prove that if the limit of $f(n)/g(n)$ exists as $n \to \infty$ and is finite, then $f \in O(g)$. Is the converse true?
(4) Compute $\sum_{i=0}^{a-1} \sum_{j=i+1}^{a} 1$.
(5) What is the temporal and spatial complexity for merging sorted lists? Explain why.
(6) Describe the merge sort algorithm and explain why its temporal complexity is $O(n \log n)$. How is it different from the naive sorting algorithm on page 17 that uses merging?
(7) Explain why linear search has temporal complexity $O(n)$ and binary search has temporal complexity $O(\log n)$.
(8) Find the remainder when 2022^{2022} is divided by 7.
(9) Show that $n! > n^n/e^n$ for all integers $n \geq 1$.
(10) If a function satisfies the recurrence $T(n) = 5(\lceil n/3 \rceil) + n^2$, $T(1) = T_1$, what can be said about the growth of $T(n)$?
(11) Starting with an empty AVL tree, show each step in adding and rebalancing for the sequence $3, 5, 2, 8, 6, 1$.
(12) Show each step in heapifying the array $[3, 5, 2, 8, 6, 1]$.
(13) T/F: If $f \in O(g)$ then $g \in O(f)$.
(14) T/F: The function $f(n) = 2^n$ satisfies $\Delta[f](n) = f(n)$.
(15) T/F: Every NP-complete problem is known to be NP-hard.