CSE 421 Midterm Review/Guide

① Start with small examples

- If there is a second variable such as k, try first with small values.
- Come up with examples dynamically that force your algorithm to its limits.

② Identify mathematical properties to rely on

Ex. Dijkstra's never resets distances after processing

A cycle-free path has length $\leq n-1$

An unsorted list has at least one adjacent inversion

③ Write a high-level algorithm first

- Ignore data structures and edge cases
- Think and write in technical English
- Approximately one sentence should suffice
- Often, correctness can be proven directly from the high-level alg.

④ Identify algorithmic techniques

- Learn the identifiable patterns of each technique.
- Is the problem self-similar? Divide & Conquer
- Are there any obvious choices the alg. must make? Greedy
- Is there a graph underlying the problem? If so, what are the vertices and what are the edges?

⑤ Use known algorithms as subroutines

- Helps make proofs clear and correct
- Helps organize thoughts and convey information

⑥ The use of bijections in proofs

- If you prove a bijection between 2 sets, then often optimazation over one is equivalent to optimization over the other
- Ex. Extensions of graph traversal or spanning tree problems