Problem 1.

(a) In class, we solved the selection problem by breaking the list into groups of 5 elements each. You can also do selection by breaking the list into groups of 7 elements each. It turns out that you need 10 comparisons to find the median of 7 elements?

  (i) Write down the recurrence for a selection algorithm based on columns with 7 elements each. (You can ignore floors and ceilings, as we did in class.)

  (ii) Solve the recurrence.

(b) (i) How may comparisons do you need to find the median of 3 elements? Give an algorithm and prove that it is optimal.

  (ii) Write down the recurrence for a selection algorithm based on columns with three elements each. (You can ignore floors and ceilings, as we did in class.)

  (iii) Solve the recurrence.

Problem 2. Show that quicksort can be implemented with worst case $\Theta(n \log n)$.

Problem 3. **Challenge Problem.** Show how to find the median of 5 numbers with only 6 comparisons.

Problem 4. Let $G = (V, E)$ be a directed graph.

  (a) Assuming that G is represented by an adjacency matrix $A[1..n, 1..n]$, give a $\Theta(n^2)$-time algorithm to compute the adjacency list representation of G. (Represent the addition of an element v to a list l using pseudocode by $l \leftarrow l \cup \{v\}$.)

  (b) Assuming that G is represented by an adjacency list $\mathrm{Adj}[1..n]$, give a $\Theta(n^2)$-time algorithm to compute the adjacency matrix of G.

Problem 5. Assume that you have an undirected, weighted graph, where some weights are positive and some are negative. You would like to find a spanning tree who sum of weights on the edges is as close to zero as possible.

  (a) Robert Prim suggests that you start at any vertex and grow a tree. Always include a new edge into the tree whose total weight makes the current sum of edge weights as close to zero as possible.

    (i) Does Prim's algorithm find a desired tree?

    (ii) If so prove it. If not, give a counterexample.

  (b) Joseph Kruskal suggests that you start with an empty tree and keep adding edges into the tree (that do not create a cycle) whose total weight of the edges is as close to zero as possible.

    (i) Does Kruskal's algorithm find a desired tree?

    (ii) If so prove it. If not, give a counterexample.