1. Pick up sticks

There are 21 sticks. Two players, A and B, alternate turns. In each turn, a player may pick up 1, 2, or 3 sticks. Player A goes first and the player who picks up the last stick wins. Use dynamic programming to determine an strategy for player A in which player A always wins. You must use dynamic programming for this problem. You might begin by playing the game with your study partners.

2. Independent set in a chain graph

An independant set in a graph is a set of vertices, no two of which are connected by an edge. We will consider only very simple graphs, for which the whole graph is a single path. So G has vertex set $V = \{v_1, v_2, ..., v_n\}$, and edge set $E = \{(v_1, v_2), (v_2, v_3), ...(v_{n-1}, v_n)\}$. For $n = 6$, an example of an independent set is $\{v_2, v_5\}$.

Now say every vertex v_i has a weight w_i. The weight of an independent set is the sum of the weights of the vertices in the set. Give the most efficient algorithm you can to find an independent set with maximum weight. Prove that your algorithm is correct, and analyze the running time. The weights might be negative; if all of the weights are negative, then the maximum weight independent set is empty.

Hint: There are some obvious ideas that don't work, so be careful.

3. Negative-weight cycles

a) Do problem 25.2-6, describing how to find negative-weight cycles.

b) Do problem 24-3, part a.

4. The shortest-path tree

The basic step of Dijkstra's algorithm adds a vertex v_j to the set S of vertices nearest to s. It chooses a v_j using the estimated distance $d(v_i) + w(e_{i,j})$ for some vertex v_i already in S. Consider coloring the edge $e_{i,j}$ when it is used to add a vertex to S. Prove that when Dijkstra's algorithm is complete, the colored edges form a directed spanning tree of G, with s as the root. Use a formal proof by induction.

5. Extra credit review problem

Do Problem 7-2, part e. You may use the results from the earlier parts of the problem without deriving them yourself.

Here is the problem, in case you don't have a 7-2 part e in your book. In the earlier parts, we proved that the running time for Quicksort can be described by the recurrence relation:

$$\mathrm{E[T(n)]} = \frac{2}{\mathrm{n}} \sum_{\mathrm{q}=0}^{\mathrm{n}-1} \mathrm{E[T(q)]} + \mathrm{O(n)} \tag{1}$$

We also showed that:

$$\sum_{k=1}^{n-1} k \lg k \le \frac{1}{2} n^2 \lg n - \frac{1}{8} n^2 \tag{2}$$

Part e asks:

Using the bound in Equation 2, show that the recurrence in Equation 1 has the solution $\mathrm{E}[\mathrm{T}(\mathrm{n})] = \mathrm{O}(\mathrm{n} \lg \mathrm{n})$. Hint: Show by substitution, that $\mathrm{E}[\mathrm{T}(\mathrm{n})] \leq \mathrm{an} \lg \mathrm{n} - \mathrm{bn}$ for some positive constants a and b.