22C : 153 **Design and Analysis of Algorithms**
Midterm Exam

1. (4 points) Solve the following recurrences using the master method, or state that the method does not apply. There is no need to provide any justification in this problem; simply state the answers.

   (a) $T(n) = 2T(n/2) + n2^{\sqrt{\log n}}$

   (b) $T(n) = 3T(n/4) + 7$

   (c) $T(n) = 4T(n/2) + n^2 - 100n + 4$

   (d) $T(n) = 2T(n/2) + 2^n$

2. (4 points) Let $S = \{p_1, \ldots, p_n\}$ be a set of n positive integers such that the sum of the integers is $3k$ for some positive integer k. We want to determine if S can be partitioned into three subsets A, B, and C such that the sum of the integers in each subset is k. Note the similarity of this problem to one of the homework problems that involved dynamic programming. (Defintion: we say that the sets A, B, and C form a partition of S if $A \cup B \cup C = S$ and $A \cap B, B \cap C, A \cap C = \emptyset$.)

   We try to solve this using dynamic programming as follows. For $1 \leq i \leq n$, let S_i denote the set of integers $\{p_1, \ldots, p_i\}$. For any triple (k_1, k_2, k_3) of integers such that $0 \leq k_1, k_2, k_3 \leq 3k$, we say that a set S_i *achieves* the triple (k_1, k_2, k_3) if there is a partition of S_i into subsets A, B, and C such that the elements of A add up to k_1, the elements of B add up to k_2, and the elements of C add up to k_3. (Assume that the elements in an empty set $\emptyset$ add up to 0.)

   (a) For $2 \leq i \leq n$, show that S_i achieves the triple (k_1, k_2, k_3) if and only if S_{i-1} achieves either $(k_1 - p_i, k_2, k_3)$, or $(k_1, k_2 - p_i, k_3)$, or $(k_1, k_2, k_3 - p_i)$.

   (b) In a few sentences, sketch an algorithm for determining if $S = S_n$ achieves (k, k, k). Your algorithm should run in time polynomial in k and n. There is no need to write elaborate psuedo-code.

3. (6 points) In an undirected graph $G = (V, E)$, we define a cut to be any partition (V_1, V_2) of V such that $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$. The *weight* of a cut is the number of edges going between V_1 and V_2. In other words, it is the number of edges (u, v) such that $\{u, v\} \cap V_1 = \{u, v\} \cap V_2 = 1$. Consider the following randomized scheme for finding a cut whose weight is close to the weight of a maximum weight cut.

   We have two sets S and T that are initially empty. We go through each vertex of the graph, we toss an unbiased coin, and add the vertex to S if the toss is a "head" or to T if the toss is a "tail". The toss for each vertex is independent of all other tosses.

   At the end of this algorithm, (S, T) is a cut. Let Y be a random variable that stands for the number of edges going between S and T. We want to show that $E[Y]$, the expected value of Y, is close to the weight of a maximum weight cut. For each edge e of the graph, define an "indicator" random variable X_e that takes a value of 1 if e goes between S and T and a value of 0 otherwise.

(a) What is the relation between Y and X_e's ?

(b) For an edge e, what is $E[X_e]$?

(c) What is $E[Y]$?

(d) How close is $E[Y]$ to the weight of a maximum weight cut?

4. (6 points) Let $G = (V, E)$ be an undirected graph, and $s, t \in V$ be two distinct vertices. Let k be the smallest number of edges that need to be removed from G in order to disconnect s from t. We want to prove Menger's theorem, which states that there are k edge-disjoint paths between s and t. A set of paths is edge-disjoint if no two paths share an edge. (Note that they can share vertices.) Hint: Construct a flow network from G, use the max-flow-min-cut theorem, argue that there is a max-flow that is integer-valued, and then proceed.

   Can there be more than k edge-disjoint paths between s and t?