INSTRUCTIONS:

1. You may directly use the algorithms introduced in lectures and assignments.
2. You always need to analyze the running time of your algorithm.
3. Reminder: you have the option of "not answering but getting 30% points" by simply writing "I choose to go for the 30% option" as your solution.

Problem 1 (12 points).

Consider this variant of subset-sum problem. You are given a set of positive integers $A = \{a_1, a_2, \cdots, a_n\}$ and a positive integer B. A subset $S \subset A$ is called feasible if the sum of the numbers in S does not exceed B, i.e., $\sum_{a_i \in S} a_i \leq B$. Design an approximation algorithm with ratio of 2 for this problem. Specifically, your algorithm should return a feasible set $S \subset A$ whose total sum is at least half as large as the total sum of the optimal solution. Prove and analyze the approximation ratio and show a tight example. Your algorithm should run in $O(n \log n)$ time.

Problem 2 (8 + 8 = 16 points).

Consider this max-cut problem: given a directed, weighted, graph $G = (V, E)$ with weight $w(e) \geq 0$ for any $e \in E$, find a cut $(S, T = V \setminus S)$ such that the total weights of cut-edges, i.e., $\sum_{e \in E(S,T)} w(e)$, is maximized, where $E(S,T) = \{(u,v) \in E \mid u \in S, v \in T\}$.

1. Design a randomized algorithm and prove that the randomized approximation ratio of it is 4.
2. Design a (deterministic) approximation algorithm using derandomization and prove that the approximation ratio of your algorithm is 4.

Problem 3 (8 + 8 = 16 points).

Consider the same max-cut problem stated in Problem 2.

1. Formulate this problem as an ILP. (*Hint:* consider using binary variable y_e for edge $e \in E$ to indicate if edge e is in $E(S,T)$, and using binary variable x_v for each $v \in V$ to indicate if v is in S.)
2. Consider the following LP + random-rounding algorithm: let $\{x_v^*\}$ and $\{y_e^*\}$ be the optimal solution of the LP-relaxation of your ILP; put v into S with probability of $1/4 + x_v^*/2$. Prove that, the randomized approximation ratio of this algorithm is 2.

Problem 4 (8 + 8 = 16 points).

Consider this clique-decomposition problem: given an undirected graph $G = (V, E)$, decide if there exists k subsets of vertices, $S_1, S_2, \cdots, S_k$, $S_i \subset V$, $1 \leq i \leq k$, such that (1), the subgraph induced by each S_i is a clique, i.e., if $u, v \in S_i$, then $(u, v) \in E$, for every $1 \leq i \leq k$, and (2), each edge must appear in at least one clique, i.e., for any $(u, v) \in E$, there must exist one subset S_i such that $u, v \in S_i$. (Note: it is possible that one vertex appears in multiple subsets.)

1. We define a pair of vertices u and v forms an *intimate-pair* if $(u, v) \in E$ and they have the same set of adjacent vertices, i.e., $\{w \in V \mid (w, u) \in E\} = \{w \in V \mid (w, v) \in E\}$. Prove that, suppose that G does not contain any intimate-pair, then $|V| \leq 2^k$. (Note: this part is a bonus question.)
2. Design a fixed-parameter tractable algorithm (w.r.t. parameter k) for above clique-decomposition problem. Prove its correctness, analyze its running time, and show that it is an FPT algorithm.