1. (i) We studied the following greedy algorithm for dense subgraph computation.

> **Algorithm**
> - Check the density of the entire graph $G = (V, E)$
> - Remove the minimum degree vertex with all its edges
> - Recheck the density of the remaining subgraph
> - Repeat (ii) and (iii)
> - Return the subgraph that has the maximum density among the ones computed

Show how to implement this algorithm to run in time $O(|V| + |E|)$.

(ii) Consider the following greedy algorithm for densest-k subgraph

> **Algorithm**
> - For $i = 1$ to $k/2$
>   - Pick the vertex with highest degree. Remove all edges incident on it.
> - End For
> - Let $U = \{v_1, v_2, \ldots, v_{k/2}\}$ be all the vertices picked.
> - Reinitiate the graph $G = (V, E)$
> - Remove any edge that is not incident on at least one vertex in U
> - For $i = 1$ to $k/2$
>   - Pick the vertex with highest degree. Remove all edges incident on it.
> - End For

Show that the above algorithm achieves an $O(\frac{n}{k})$ approximation for densest k subgraph.

*Points 10+20*

2. (i) Suppose the size of each set in unweighted set cover instance is at most $d$. Show that the greedy algorithm achieves an $(\ln d + 1)$ approximation.

*Hint.* Suppose $OPT = k$. Show that after the $ith$ set is chosen by the greedy algorithm, at most $\left(1 - \frac{1}{k}\right)n$ elements are left to be covered.
(ii) The tight example shown for unweighted set cover in the class when converted to a vertex cover instance results in a multigraph. Give an example of a simple graph, where the greedy algorithm has logarithmic approximation bound.

(iii) Given an undirected graph \( G = (V, E) \), a dominating set of a graph is a subset \( S \subseteq V \) such that each vertex either belongs to \( S \) or has a neighbor in \( S \). The minimum dominating set problem finds a dominating set of minimum size. Give a reduction from Set Cover to minimum dominating set problem.

**Points 30+20+10**

3. For bipartite graphs, size of the minimum vertex cover is same as the maximum matching. Give an algorithm to compute vertex cover exactly in bipartite graphs.

*Hint. Use maxflow computation in a newly constructed graph.*

**Points 30**

4. (i) Given a ground set \( V \), a function \( f : 2^V \to \mathbb{R}_+ \) is said to be submodular if either of the two conditions hold

(a) For all subsets \( A, B \subseteq V \), \( f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \)

(b) For all \( A \subseteq B \), and \( v \not\in B \), \( f(A + v) - f(A) \geq f(B + v) - f(B) \)

Show that the two conditions are equivalent.

(ii) Let \( f : 2^V \to \mathbb{R}_+ \) be a non-negative submodular function on a set \( V \). Let \( A \subset B \subseteq V \). Prove the followings.

(a) If \( f(A) < f(B) \) then there exists an element \( v \in B \setminus A \) such that \( f(A + v) - f(A) > 0 \).

More generally, there exists an element \( v \in B \setminus A \) such that \( f(A + v) - f(A) \geq \frac{f(B) - f(A)}{|B \setminus A|} \).

(b) If \( f(B) < f(A) \) then there exists an element \( v \in B \setminus A \) such that \( f(B - v) - f(B) > 0 \).

More generally, there exists an element \( v \in B \setminus A \) such that \( f(B - v) - f(B) \geq \frac{f(A) - f(B)}{|B \setminus A|} \).

**Points 10+20**