Problem 3. Routing Packets from N to M [16 points]

There are a set of computers N here at NUS, and a set of computers M at MIT (in the USA). Connecting the computers in N and the computers in M is a network. We want to know how many different routes there are to send packets from N to M, so that we can reliable send data from here to MIT, even if some of the links have been compromised by an attacker.

Formally, you are a given a graph $G = (V, E)$ with a designated subset $N \subseteq V$ and $M \subseteq V$, where N and M are disjoint. We say that a path is a NM-path if it starts at a node in N and ends at a node in M. Two NM-paths p_1 and p_2 are said to be *edge disjoint* if there is no edge $e \in E$ that is in both path p_1 and path p_2.

Give an efficient algorithm to find a maximum-sized set of edge disjoint paths from N to M. (The output of the algorithm should be a set of edge disjoint paths.) Give the running time of your algorithm, prove that your algorithm is correct, and show that it finds the *maximum* number of such paths. *Hint: Remember the flow decomposition lemma.*

Problem 4. (Edge-disjoint Path Covering) [25 points]

Assume you are given a connected graph $G = (V, E)$ consisting of n nodes and m edges. Prove the following fact:

> There exists a set of at most $\lfloor n/2 \rfloor$ edge-disjoint paths P_1, P_2, P_2, P_k, where $k \leq \lfloor n/2 \rfloor$, such that every edge in E appears in exactly one path P_j.

(Recall that two paths P_i and P_j are edge disjoint if they do not share any edges.) We say that the graph G is ***covered*** by k edge-disjoint paths.

Hint: Imagine adding some fake edges, to be removed later, in order to create nodes with even degree.

Problem 4. SecretNets Corporation Incorporated, Ltd. [26 points]

SecretNets Corp. has built an overlay network for tranporting data across the internet. The SecretNet consists of a directed graph $G = (V, E)$ with a set of entry nodes $e_1, e_2, \ldots, e_k$ and a set of exit nodes $x_1, x_2, \ldots, x_k$. SecretNets makes the following guarantee to its clients: "Your packets will be routed securely through the network, and they will never share a node or an edge in the network with a competitor." This ensures that there is no way that anyone can spy on their packets!

Problem 4.a. [8 points] The engineer who originally designed the SecretNet just quit. Your job, as the replacement, is to find the maximum number of clients that the SecretNet can support.

- Each client is assigned some entrypoint e_i and some exit point x_j. (Any combination of entry point and exit point is allowed.)
- Each client is assigned a path through the network from e_i to x_j.
- No two client paths intersect at any **nodes or edges**.

Give an efficient algorithm for finding the maximum number of clients that the SecretNetwork can support.

Problem 4. SecretNets Corporation Incorporated, Ltd. [26 points]

SecretNets Corp. has built an overlay network for tranporting data across the internet. The SecretNet consists of a directed graph $G = (V, E)$ with a set of entry nodes $e_1, e_2, \ldots, e_k$ and a set of exit nodes $x_1, x_2, \ldots, x_k$. SecretNets makes the following guarantee to its clients: "Your packets will be routed securely through the network, and they will never share a node or an edge in the network with a competitor." This ensures that there is no way that anyone can spy on their packets!

Problem 4.a. [8 points] The engineer who originally designed the SecretNet just quit. Your job, as the replacement, is to find the maximum number of clients that the SecretNet can support.

- Each client is assigned some entrypoint e_i and some exit point x_j. (Any combination of entry point and exit point is allowed.)
- Each client is assigned a path through the network from e_i to x_j.
- No two client paths intersect at any **nodes or edges**.

Give an efficient algorithm for finding the maximum number of clients that the SecretNetwork can support.

Problem 4.b. [18 points] It turns out that there are only d clients, at the moment, fewer than the maximum number that your network can support. Each link in the network has a cost to use it, and you want to find the d disjoint paths that are cheapest. That is:

- Assume each edge in the SecretNetwork $e \in E$ has a bandwidth cost $b(e)$.
- Your algorithm should output d paths $p_1, p_2, \ldots, p_d$ that are entirely disjoint, that is, they do not intersect at either the nodes or edges.
- The cost of a path p_j is the sum of the edge costs, i.e., $\sum_{(v,w) \in p_j} b(v, w)$.
- Your algorithm should minimize the total cost, i.e., $\sum_{j=1,\ldots,d} cost(p_j)$.

Give an efficient algorithm, and prove that it satisfies the requirements.

(For partial credit, you may instead give an algorithm that finds client paths that do not intersect at any ***edge*** *in the SecretNetwork.)*

Problem 1. Warehouse Management [20 points]

You have been placed in charge of the delivery infrastructure for an on-line grocery store. Your job is to decide where to situate warehouses in order to minimize the costs of operating the warehouse and shipping the groceries to the customers. You have information on all your customers, including their location and the amount of goods that they want. Each customer must receive all the required groceries, and may receive shipments from one or more warehouses. In more detail, for this problem, you are given:

- A set of customers $c_1, \ldots, c_n$, where each c_i represents the location of customer c_i. (Each location is a set of coordinates in Euclidean space.)
- The demand d_i for each customer c_i. That is, customer c_i requires d_i kilograms of groceries.
- A set of possible locations to open warehouses $w_1, \ldots, w_k$. Each w_j repesents the location of warehouse site w_j.
- The cost v_j of operating each warehouse w_j. That is, to operate a warehouse at location w_j costs w_j dollars.
- The cost Z for shipping 1 kilogram of groceries 1 kilometer. That is, if you are shipping x kilograms of groceries to a customer that is y kilometers from the warehouse, then it costs xyZ dollars.

Your job is to formulate this problem as an integer linear program. (You do not have to solve the problem. You simply have to write the appropriate integer linear program.) Your integer linear program does not have to be in standard form, however it should only use constructs that we have already shown can be transformed into standard form.

Please write your answer on the next page.

Variables: *(For each variable, explain its intuitive meaning.)*

Objective:

Constraints: *(For each constraint, first describe in English the goal of the constraint. Then provide the mathematically precise constraint.)*