**CS 514**

**report2.txt**

1. You are given a directed graph that does not have any cycles. Give a polynomial time algorithm that outputs the vertices in an order satisfying that if (a, b) is an edge, then a is output before b.

HINT: Find a greedy algorithm for this problem.

Solution: We give a greedy algorithm. First we claim that a directed acyclic graph must have a vertex with no in-edges. To see this, suppose for the sake of contradiction that there is a directed graph on n vertices, with no cycles and yet every vertex has an outgoing edge. Then let v0, v1, . . . , vn be a sequence in the graph such that (vi+1, vi) is always an edge. Namely, we walk backwards along edges of the graph for n steps. Since every vertex has an incoming edge, we can find such a long sequence by repeatedly picking an incoming edge of vi . By the pigeonhole principle, some vertex must repeat in this sequence. But then, just as we saw in class, we obtain a cycle in the directed graph, this is a contradiction. Then the greedy algorithm is as follows:

(a) Find a vertex a with no incoming edges, which must exist since the graph is acyclic.

(b) Output a, and delete it from the graph, to obtain a new acyclic graph.

(c) Repeat this until no more vertices are left.

The algorithm can be implemented to run in time O(n(m + n)). First make a pass on the adjacency list representation to add a list of all the incoming edges. Then in each step we need to scan all the vertices to find a vertex with no incoming edges, which takes time O(n). To prove that the algorithm is correct, suppose (a, b) is an edge of the graph. Then we claim that a is output before b. Indeed, this is because if b is about to be output, then b cannot have any incoming edges, so a must have been output first.

2. A table of characters and corresponding frequencies is provided below. Next to the table, draw a valid Huffman encoding tree. Then fill out the code map table with each character’s encoding according to your Huffman encoding tree.

Character – Frequency

A – 5

B – 2

C – 8

D – 7

E – 3

F – 7

G – 1

Solution – We create a minimum priority queue and keep iteratively combining the least 2 nodes and updating the min priority queue. We get

In sheet

3. (10 points) Design a dynamic programming algorithm for longest increasing subsequence.

(10 points) Prove its correctness and analyze its running time.

Let the input have size n, we build a size n table A by,

j \* (i) = argmax{A[j] such that j < i and aj ≤ ai}

A[i] = 1 + A[j \*(i)]

j \*(i) is set to zero if no choice for j exists (for example on the first element of the list).

The proof is based on the fact that A[j] holds the length of the longest increasing subsequence that ends on aj . This is proved by induction, where the base case is easy. For the inductive step, consider some index i, we set A[i] = 1 + A[j \* (i)], and we are guaranteed that j \* (i) maximizes A[j] among all possible positions j that could come before i in an increasing subsequence.

Thus we are ensured that A[i] holds the proper value. The running time is O(n^2 ).

4. A ship can carry a maximum weight of **W tons** and has a maximum volume of **V cubic meters**. It carries ki types of materials whose densities are given by di tons/cubic meters, maximum available volumes of vi cubic meters, and revenues ri per cubic meter where 1 ≤ i ≤ k.

Formulate a linear program to determine the optimal volumes of each type to maximize the total revenue.

5. Mail Delivery Problem: You are given a map of a city which is in the form of directed graph (V, E) where each edge E has a weight w(e) that represents the distance of that road segment. You are also given a subset of vertices S (places where the mail should be delivered) and a starting vertex (the post office) U. You want to find a minimum length tour that starts from u, visits all vertices in S (and possibly some others) and returns to u.

(a) (10 points) Define a decision problem that corresponds to the Mail Delivery Problem and show that it is in NP.

(b) (10 points) Show that the decision problem you introduced is NP-Complete by reducing the directed Hamiltonian Cycle (dHamCycle) problem to it.

**(a)**

The Mail Delivery Problem is a decision problem that corresponds to finding the minimum-length tour that starts from the post office, visits all the required delivery locations, and returns to the post office. This problem is in NP because, given a proposed tour, it can be verified in polynomial time whether it is a valid tour that satisfies the requirements (i.e., it starts and ends at the post office, visits all the required delivery locations, and has the minimum length).

**(b)**

To show that the Mail Delivery Problem is NP-complete, we can reduce the directed Hamiltonian Cycle (dHamCycle) problem to it. The dHamCycle problem is the problem of determining whether a given directed graph contains a Hamiltonian cycle (a cycle that passes through every vertex exactly once) or not. This problem is known to be NP-complete.

To reduce dHamCycle to the Mail Delivery Problem, we can start with a given directed graph G and a proposed Hamiltonian cycle C in G. We construct a new instance of the Mail Delivery Problem as follows:

* The map of the city is the given directed graph G.
* The starting vertex is the first vertex in the proposed Hamiltonian cycle C.
* The subset of vertices where the mail should be delivered is the set of vertices in the Hamiltonian cycle C.
* The minimum-length tour is the proposed Hamiltonian cycle C itself.

If the proposed Hamiltonian cycle C is a valid cycle in G, then it will also be a valid tour in the constructed instance of the Mail Delivery Problem, and it will have the minimum length (because it is the only tour that visits all the required delivery locations). On the other hand, if the proposed Hamiltonian cycle C is not a valid cycle in G, then it will not be a valid tour in the constructed instance of the Mail Delivery Problem, because it will not be possible to visit all the required delivery locations. Therefore, in either case, the proposed Hamiltonian cycle C will be a correct answer to the constructed instance of the Mail Delivery Problem if and only if it is a correct answer to the dHamCycle problem.

Since the reduction can be performed in polynomial time, this shows that the Mail Delivery Problem is NP-complete.

**Explanation:**

To show that the Mail Delivery Problem is NP-complete, we need to reduce the directed Hamiltonian Cycle (dHamCycle) problem to it. The dHamCycle problem is the problem of determining whether a given directed graph contains a Hamiltonian cycle (a cycle that passes through every vertex exactly once) or not. This problem is known to be NP-complete, which means that any problem that is NP-hard (i.e., at least as hard as any NP-complete problem) and in NP can be reduced to it.

To reduce dHamCycle to the Mail Delivery Problem, we can start with a given directed graph G and a proposed Hamiltonian cycle C in G. We construct a new instance of the Mail Delivery Problem as follows:

* The map of the city is the given directed graph G. This means that the set of vertices V and the set of edges E in the constructed instance of the Mail Delivery Problem are the same as those in the given graph G.
* The starting vertex is the first vertex in the proposed Hamiltonian cycle C. This means that the starting vertex u in the constructed instance of the Mail Delivery Problem is the same as the first vertex in the proposed Hamiltonian cycle C.
* The subset of vertices where the mail should be delivered is the set of vertices in the Hamiltonian cycle C. This means that the subset of vertices S in the constructed instance of the Mail Delivery Problem is the same as the set of vertices in the proposed Hamiltonian cycle C.
* The minimum-length tour is the proposed Hamiltonian cycle C itself. This means that the proposed Hamiltonian cycle C is also a proposed tour in the constructed instance of the Mail Delivery Problem.

Now, consider the proposed Hamiltonian cycle C. If it is a valid cycle in G, then it will also be a valid tour in the constructed instance of the Mail Delivery Problem, because it starts and ends at the post office (the first vertex in the cycle), visits all the required delivery locations (the other vertices in the cycle), and has the minimum length (because it is the only tour that visits all the required delivery locations). On the other hand, if the proposed Hamiltonian cycle C is not a valid cycle in G, then it will not be a valid tour in the constructed instance of the Mail Delivery Problem, because it will not be possible to visit all the required delivery locations. Therefore, in either case, the proposed Hamiltonian cycle C will be a correct answer to the constructed instance of the Mail Delivery Problem if and only if it is a correct answer to the dHamCycle problem.

Since the reduction can be performed in polynomial time, this shows that the Mail Delivery Problem is NP-complete.