Underlying Concepts

- Shortest Path/ Graph Processing
- Greedy – (Proving optimality of greedy *)
- Minimum spanning Tree – (cycles and cuts)
- Huffman Coding (Exercise)
- Dynamic Programming (Extension of Bellman-ford)
Problem 1: Given an edge-weighted undirected connected chain-graph \( G = (V,E) \), all vertices having degree 2, except two endpoints which have degree 1 (there is no cycle). Design an algorithm that preprocesses the graph in linear time and can return the distance of the shortest path between any two vertices in constant time (i.e., the \( O(|V|) \) preprocessing enables return of the shortest-path distance between any two vertices in \( O(1) \) time).

- Graph is line like.
- Preprocessing(graph, preprocessed){
  // Takes linear time (\( O(V) \)).
}
- ShortestDistance(int u, int v, preprocessed){
  // return shortest-distance between u and v in constant time.
}
Problem 1 continue

- Shortest path distance between B and C in this diagram?
- Can I find this in constant time if I store some Information about B and C already?
Problem 2: Recall interval scheduling problem from class (the interval scheduling problem is to find a largest compatible set - a set of non-overlapping intervals of maximum size). Suppose that instead of always selecting the first activity to finish, we select the last activity to start that is compatible with all previously selected activities. Describe how this approach is a greedy algorithm, and prove that it yields an optimal solution.

- How approach is greedy? Definition of greedy?
- Prove it yields optimal solution?
Problem 3: Let $T$ be an MST of a weighted, undirected graph $G$. Given a connected sub-graph $H$ of $G$, show that $T \cap H$ is contained in some MST of $H$.

- Try to prove by contradiction,
  Assume an edge $e \in T \cap H$ and $e \notin \text{MST}(H)$
- Not that straight-forward, make MSTs using Kruskal’s Algorithms.
Problem 4: Second best minimum spanning tree?

- For this example try to make MST, is it unique?
  \[ \text{Cost(MST)} = 26 \]
- Now find out second best minimum spanning tree, more than one possible??
  \[ \text{cost(second best mst)} = 27 \]

Algorithm to find second best mst:
1. Find MST
2. Some modification in MST (hint: b and c part)
Problem 5: Given a graph $G = (V, E)$, a subset $S \subseteq V$ of vertices is said to be a vertex cover of $G$ if for every edge $(u, v) \in E$, at least one of $u$, $v$ belongs to the subset $S$. A minimum vertex cover of $G$ is a vertex cover with minimum cardinality (i.e., smallest vertex cover).

Design an algorithm to find a minimum vertex cover of a given tree $T = (V, E)$.

- What is minimum vertex cover?
- Minimum vertex cover of Tree
Problem 5 Continue

Greedy Algorithm

Suggestion 1: Pick vertex with maximum degree
Contradictory example:
This approach will give vertex cover with cardinality 5 \{A, B, C, D, E\}
minimum vertex cover is of cardinality 4 \{B, C, D, E\}

Suggestion 2: Pick Alternate layers
(Decide which set of alternate layers give minimum cardinality)
Contradictory example:
This approach will give vertex cover with cardinality 3
either \{A, E, F\} or \{B, C, D\}
minimum vertex cover is of cardinality 2 \{A, D\}
Any Questions?