263 final review

* What does P stand for
  + Polynomial
* A greedy algorithm
  + Selects choices based on short term advantages
* An adjacency list contains
  + A list of all vertices and their adjacent vertices
* A lookup in adjacency list is faster than adjacency matrix
  + false
* The TSP problem falls under
  + NP complete
* The space requirement for an adjacency matrix is
  + |V| \* |V|
* The cost of a path
  + Can be found by summing all the cost of the edges in the path
* The max flow problem is the same as
  + The minimum cut problem
* DAG stands for
  + Directed acyclic graph
* Space requirement for an adjacency list is
  + |V| + |E|
* Ford Fulkerson algorithm for max flow
  + Utilizes 3 graphs to solve the problem
* Most graphs that represent real world data
  + Are sparse
* Which can be implemented with dynamic programming
  + Fibonacci and knapsack
* If the edges in the graph are ordered
  + The graph is directed
* Which graph has the most edges
  + A dense graph
* An undirected graph is connected if
  + There is a path from every vertex to every other vertex
* The knapsack problem when implemented using dynamic programming is
  + NP complete
* A simple path
  + All vertices are distinct except the first and last
* A graph can be represented by
  + An adjacency list or adjacency matrix
* A directed graph is acyclic if
  + It has no cycles
* A graph that is sparse is best represented by
  + An adjacency list
* In an undirected graph, the edge (u,v) is equivalent to
  + The edge (v,u)
* The TSP problems is most similar to
  + The Hamiltonian Circuit
* The max depth of a binary tree
  + N - 1
* An NP complete problem can be reduced to what other type of problem
  + NP complete problems
* The indegree for a node V in a directed graph is defined by
  + The number of edges (u,v)
* A graph is defined by
  + Vertices and edges
* A tree with 2 children for each node is called
  + Binary tree
* Dynamic programming takes advantage of a technique
  + That solves and records subproblems
* A topologic sort requires the graph
  + To be acyclic / no cycles
* Dynamic programming takes advantage of
  + Solves and records subproblems
* The indegree for a node V in a directed graph is defined by
  + The number of edges (u, v)
* An NP complete problem can be reduced to
  + Other NP complete problems
* The max depth of a binary tree is
  + N - 1
* A graph is defined by
  + Vertices and Edges
* Ford Fulkerson uses
  + Up to 3 graphs to solve the problem
* A thee with two children for each node is
  + A binary tree
* Minimum cut I closest to
  + Max flow
* Topological sort requires the graph
  + To be acyclic
* In an undirected graph the edge (u, v) is equivalent to
  + The edge (v, u)
* Another name for an NP complete problem
  + A certificate
* Set of classes that can be reduced to one another
  + NP complete
* The halting problem takes what as input
  + Another program
* The halting problem
  + Cannot be solved
* A problem that is NP complete
  + Has a solution that can be verified in polynomial time
* P = NP
  + Is unknown because you can’t prove it
* The knapsack problem is
  + NP complete – not known as the ham circuit – known as the TSP
* Verifying a solution for an NP-complete problem takes
  + Polynomial time
* The TSP problem is also known as
  + The minimum cost Hamiltonian circuit