**3**

dags: a directed acyclic graph is a directed graph with no directed cycles. In other words formed by a collection of vertices and directed edges, each edge connecting one vertex to another, such that there is no way to start at some vertex v and follow a sequence of edges that loop back to V again.

dfs on dags:if dfs reveals a back edge then it is not a dags

topological sorting: order the vertices one after another in such way that each edge goes from an earlier vertex to a later one. So that all preexisting constraints are satisfied.

 strong connected: a graph can be decomposed into several connected components. A new component is created when the dfs restarts. To be connected in a directed graph the two nodes must have a path from v to u and u to v.

**from 4**

shortest path: using breadth-first search it is possible to find the shortest path from point a to point b.

dijkstra's:checks all the nodes around edge e for their length and based upon that takes the lowest length and removes the edge it was at.

negative weights:dijkstra’s algorithm will not work if there are negative edges. A distance that has a negative value

negative cost cycles: if a cycle forms with negative numbers.

bellman ford algo: as long as no negative cycle is reachable from the vertex.

min spanning tree:a tree that reaches all nodes once and takes the shortest path to get to all these nodes.

**from 5**

kruskals: an algorithm to find the minimum spanning tree. It creates a set of tree where each vertex in the graph is a second tree. Then it creates a set containing all the edges in the graph. Then while the set is not empty and the tree isn’t spanning it removes an edge with the least weight from the set. If the edge connects two tree then it is added to the set of trees and the two separate trees merge, if not remove the edge.

coin changing: using dynamic programing returns the least amount of coins used to return that change. Can be set for different values to make more efficient.

Huffman:the encoding of charactes based on their usage. If it is heavier used it will have a shorter encoding, less will have longer. See hw.

perfect matching on a tree: a set of edges that touches each node exactly once. A set of edges without common vertices.

fractional knapsack: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible

**from 6**

knapsack with and without repetition: with: turns It into finding the longest path by making a dag and having the value be represented in the length. If an item is too heavy for the sack, then it cannot be reached. The graphs connect to themselves and have a weight aswell.

Without:restricted amount of times that a certain item can be used.

longest common subsequence/substring:based on two separate things find the subsequence in the two items that is the longest. See 6.3

longest increasing subsequence/substring: a subset of number taken in order in which the numbers are getting larger. Goes until the numbers stop getting larger.