Practice problems

Dynamic Programming
Dynamic programming (2D design)

We are given a list of $n$ items with sizes $s_1, s_2, \ldots, s_n$. A sequential bin packing of these items is an assignment of items to bins, such that in each bins the items are consecutive. (That is, each bin has items $s_i, s_{i+1}, \ldots, s_j$ for some indices $i < j$.) Bins have unbounded capacities. The load of a bin is the sum of the elements in it. Give an algorithm that determines a sequential packing of $n$ items into $k$ bins for which the maximum load of a bin is minimized. Analyze the time-complexity and space-complexity.

Problem (shortest paths)

- Suppose that all edge weights of a weighed connected graph are either 1 or 2
- Give an $O(n+m)$-time algorithm to solve the single-source shortest-paths problem in such graphs
Single source shortest path

Dijkstra's single-source shortest-path uses a set $C$ (the "cloud") of vertices that initially contains only the source $s$ and that eventually includes all the vertices of the graph $(V,E)$. Vertices are added to $C$ one at a time, as explained in class. Let $f(v)$ the number of times that the label $D[v]$ of a vertex $v$ in $V - C$ changes due to an edge relaxation. Answer each of the following questions (provide an intuitive explanation for each of your answers)

- Can the label $D[v]$ of a vertex $v$ in $V - C$ ever get smaller than the cost of a shortest $s$-to-$v$ path in the graph $G$?
- Can $f(v)$ become greater than the degree of $v$? (recall that the degree of a node is the number of edges incident to it)
- Can $f(v)$ be less than the degree of $v$?

Minimum spanning tree

Consider the Generic-MST algorithm in class and presented below.

Algorithm Generic-MST $(G : \text{weighted graph})$
1. let $A \leftarrow \emptyset$
2. while $A$ does not form a spanning tree for $G$ do
3. find a cut $C$ that respects $A$
4. let edge $(u,v)$ be a light edge crossing $C$
5. let $A \leftarrow A \cup \{(u,v)\}$
6. return $A$

Prove by induction on the size of $A$ that lines 3-5 cannot introduce cycles in $A$. You can assume for simplicity that $A$ is a single tree (instead of a forest). This is simple proof, but you must write all the steps to get full credit.
MST (proof)

Let $G = (V,E)$ be a weighted graph. Prove that if all edge weights in $G$ are distinct, then $G$ has a unique minimum spanning tree. Note: For full credit you have to give a formal proof. Arguments based on Kruskal's or Prim's algorithm will not be accepted.

Here is the beginning of the proof:

We assume all edge weights in $G$ are distinct. For sake of contradiction, suppose that $G$ has two minimum spanning trees $T_1$ and $T_2$. Since $T_1 \neq T_2$ they must differ in at least one edge. Let $e \ldots$