CSE 101, Winter 2018

HW1 review session
Week 1

January 8 - January 15
Important

- Look through **Additional Resources** (more practice problems and reading material) at the end of the website:

  http://vlsicad.ucsd.edu/courses/cse101-w18/
Question 1

- You are given two functions $f(n)$ and $g(n)$. Determine whether $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, or $f(n) = \Theta(g(n))$

1. Let $\lim_{n \to \infty} f(n) / g(n) = C$
   - i. If $C < \infty$, then $f(n) = O(g(n))$
   - ii. If $C > 0$, then $f(n) = \Omega(g(n))$
   - iii. If $0 < C < \infty$, then $f(n) = \Theta(g(n))$

   Using l'hopital's Rule and the definition of the limit may be useful

2. Use definitions of $O$, $\Omega$, and $\Theta$ classes

3. Let $\lim_{n \to \infty} a(n) = 0$. If $0 \leq b(n) \leq a(n)$ for all $n > n_0$, then $\lim_{n \to \infty} b(n) = 0$
Question 2  (DPV 3.3, 3.4)

Previsit and postvisit numbers for vertices are obtained during the DFS algorithm.

1. Previsit (“pre number”) of a vertex $v$ is the time when this vertex was discovered (the time when DFS enters the vertex $v$ “for the first time”)
2. Postvisit (“post number”) of a vertex is the time of the final departure from this vertex (the time when DFS exits the vertex $v$)

Time starts at 1 and increases by 1 every time we visit a vertex or exit from a vertex.
Question 2 (DPV 3.3, 3.4)

Topological ordering of a graph is an ordering of its vertices $v_1, v_2, \ldots, v_n$ in such a way that there is an edge $(v_i, v_j)$ only if $i < j$.

- Topological order of a directed graph $G$ exists only if graph $G$ is acyclic
- Any directed acyclic graph has at least one topological order of its vertices
- Can be found in a linear time using DFS algorithm

How many topological orderings exist in a graph with $n$ vertices and 0 edges?
Question 2 (DPV 3.3, 3.4)

Topological ordering of a graph is an ordering of its vertices $v_1, v_2, \ldots, v_n$ in a such way, that there is an edge $(v_i, v_j)$ only if $i < j$.

- Topological order of a directed graph $G$ exists only if graph $G$ is acyclic
- Any directed acyclic graph has at least one topological order of its vertices
- Can be found in a linear time using DFS algorithm

How many topological orderings exist in a graph with $n$ vertices and 0 edges?

Answer: $n!$
Question 2 (DPV 3.3, 3.4)

Strongly connected components (you may refer to the discussion section slides for more examples and definitions)

- $G_R$ is a reversed graph (keep all the vertices and reverse all edges in an original graph $G$)
- Metagraph: Find SCC of a graph $G$, assign a meta-node to each of SCC, add an edge from a meta-node $v$ to a meta-node $u$ if there is an edge from a SCC corresponding to $v$ to a SCC corresponding to $u$ in the original graph $G$. 
Question 3 (DPV 3.7)

- There are many “equivalent” definitions of a bipartite graph. Choose one of the definitions and try to implement a linear time algorithm. (Some definitions may simplify your search for a solution)

- The prove of a statement “A is true if and only if B is true” should consist of two parts:
  - Let’s assume that A is true. Now, we will show that B is also true... (the assumption that A is true should be used)
  - Let’s assume that B is true. Now, we will show that A is also true... (the assumption that B is true should be used)

- **Providing examples is not a proof**, however, they may help you to see some patterns and give some ideas of how the proof should look like
Question 4  (DPV 3.10)

- We have discussed postvisit and previsit numbers in Question 2.
- Stack is a container of some objects. Objects are inserted and removed in accordance with the “Last in First out” principle.
Question 5 (DPV 3.25)

- Assume, we have an edge \((v, u)\) and we know the value for \(\text{cost}[u]\). Can we use this value in any way to find/estimate the value for \(\text{cost}[v]\)?
- If the answer is yes, then which order of vertices should we have to find/estimate \(\text{cost}[v]\) using \(\text{cost}[u]\)?
- The above algorithm works only for acyclic directed graphs. Can we transform an arbitrary directed graph into some acyclic directed graph in such a way that the information for all \(\text{cost}[v]\) in an original graph is not lost and can be easily computed?
Question 6 (DPV 3.23)

- Assume, we have an edge \((v, t)\) and we know the number of paths from \(s\) to \(v\). Can we use this value in any way to find/estimate the number of paths from \(s\) to \(t\)?
- If the answer is yes, then which order of vertices should we have to find/estimate the number of paths from \(s\) to \(t\) using the number of paths from \(v\) to \(t\)?
Question 7

- Can be represented as a graph, where each point will be assigned a unique vertex
- The graph will be undirected (why?)
- The graph will be full, i.e. there is an edge between any pair of vertices (why?)
- How does the answer given by a greedy algorithm relate to the optimal answer? Does this relation depend on the input size, i.e. n?