COMP 3804 — Assignment 3

Due: Monday March 25, 23:59.

Assignment Policy:

- Your assignment must be submitted as one single PDF file through Brightspace.

> Use the following format to name your file:
>
> `LastName_StudentId_a3.pdf`

- **Late assignments will not be accepted. I will not reply to emails of the type "my internet connection broke down at 23:57" or "my scanner stopped working at 23:58", or "my dog ate my laptop charger".**
- You are encouraged to collaborate on assignments, but at the level of discussion only. When writing your solutions, you must do so in your own words.
- Past experience has shown conclusively that those who do not put adequate effort into the assignments do not learn the material and have a probability near 1 of doing poorly on the exams.
- When writing your solutions, you must follow the guidelines below.
  - You must justify your answers.
  - The answers should be concise, clear and neat.
  - When presenting proofs, every step should be justified.

```
Algorithm DFS(G):
for each vertex v
do visited(v) = false
endfor;
clock = 1;
for each vertex v
do if visited(v) = false
    then EXPLORE(v)
    endif
endfor
```

```
Algorithm EXPLORE(v):
visited(v) = true;
pre(v) = clock;
clock = clock + 1;
for each edge (v, u)
do if visited(u) = false
    then EXPLORE(u)
    endif
endfor;
post(v) = clock;
clock = clock + 1
```

Question 1: Write your name and student number.

Question 2: Let $G = (V, E)$ be a directed graph. After algorithm DFS(G) has terminated, each vertex has a *pre*- and *post*-number. Let u and v be two distinct vertices in V. Assume that both of the following are true:

- There is a directed path in G from u to v.
- $pre(u) < pre(v)$.

Professor Lionel Messi claims that, in the DFS-forest, v must be in the subtree of u.

Is Professor Messi's claim correct? As always, justify your answer.

Question 3: Give an example of a directed graph $G = (V, E)$ that contains a vertex v having the following properties:

- v has at least one incoming edge and at least one outgoing edge.
- Consider the DFS-forest after algorithm DFS(G) has terminated. This forest contains a tree containing only the vertex v.

Question 4: Let $G = (V, E)$ be a directed graph that is given to you using adjacency lists. The vertices of V are numbered arbitrarily as $v_1, v_2, \ldots, v_n$.

For each vertex u of V, let $R(u)$ be the set of all vertices v such that there exists a directed path in G from u to v, and let $Min(u)$ be the smallest index of all vertices in $R(u)$. Thus,

$$Min(u) = \min\{i : v_i \in R(u)\}.$$

Give an algorithm that computes, in $O(|V| + |E|)$ time, the value of $Min(u)$ for all vertices u in V. As always, justify your answer.
Hint: Consider the graph G' obtained from G by reversing the direction of all edges in G. Algorithm DFS is useful for this question.

Question 5: Let $G = (V, E)$ be a connected undirected graph in which each edge $\{u, v\}$ has a weight $wt(u, v)$. Consider the following algorithm:

- Let $G' = G$.
- While G' contains a cycle:
  - Let C be an arbitrary cycle in G'.
  - Let e be an edge of C whose weight is maximum.
  - Delete the edge e from G'.
- Return the graph G'.

Prove that the output of this algorithm is a minimum spanning tree of the input graph G.

Question 6: Let $G = (V, E)$ be a directed graph in which each edge (u, v) has a weight $wt(u, v)$. Let s be an arbitrary source vertex. Assume that the following are true:

- Edges that leave s may have negative weights.
- All other edges have positive weights.
- There is no cycle with negative weight.

Professor Justin Bieber claims that Dijkstra's algorithm correctly computes, for each vertex v, the length of a shortest path from s to v.

Is Professor Bieber's claim correct? As always, justify your answer.

Question 7: Let $S = (a_1, a_2, \ldots, a_n)$ be a sequence of n numbers. A subsequence T of S is called *awesome*, if for every i with $1 \leq i \leq n-1$, a_i and a_{i+1} are not both in T. In other words, whenever a number is in T, none of its neighbors in S is in T. The weight of the subsequence T is the sum of all numbers in T.

Give a dynamic programming algorithm that computes, in $O(n)$ time, the maximum weight of any awesome subsequence of T.

As always, justify your answer. Follow the three dynamic programming steps that we have seen in class.

Question 8: Tyler is not only your friendly TA, he is also the CEO of *Tuttle Enterprises*. This company buys long copper wires, cuts them into subwires, and then sells the subwires. Tuttle Enterprises only buys copper wires having integer lengths and cuts them such that each subwire has an integer length.

Let n be a large integer and let $p_1, p_2, \ldots, p_n$ be a sequence of positive numbers. For each i with $1 \leq i \leq n$, Tuttle Enterprises sells a subwire of length i for p_i dollars.

Consider a copper wire of length n. Give a dynamic programming algorithm that computes, in $O(n^2)$ time, the maximum revenue that can be obtained by cutting the length-n wire into subwires.

As always, justify your answer. Follow the three dynamic programming steps that we have seen in class.

For example, let $n = 4$. Here are the different options to cut a length-4 wire:

- The wire is not cut. Then the revenue is p_4.
- The wire is cut into one subwire of length 1 and one subwire of length 3. Then the revenue is $p_1 + p_3$.
- The wire is cut into two subwires of length 1 and one subwire of length 2. Then the revenue is $2 \cdot p_1 + p_2$.
- The wire is cut into two subwires of length 2. Then the revenue is $2 \cdot p_2$.

- The wire is cut into four subwires of length 1. Then the revenue is $4 \cdot p_1$.