CSCI 411 - Advanced Algorithms
Assignment 1

Solutions to the written portion of this assignment should be submitted via PDF to Canvas. Make sure to justify your answers. C++ code should be submitted on INGInious. Both parts of the assignment are due before **September 14th at 11:59 pm**.

There may be time in class to discuss these problems in small groups and I highly encourage you to collaborate with one another outside of class. However, you must write up your own solutions **independently** of one another. Feel free to communicate via Discord and to post questions on the appropriate forum in Canvas. Do not post solutions. Also, please include a list of the people you work with at the top of your submission.

Written Problems

1. (10 pts) Sort the following functions in terms of asymptotic growth from smallest to largest. In particular, the resulting order $f_1, \ldots, f_{12}$ should be such that $f_1 = O(f_2)$, $f_2 = O(f_3)$, and so on. Identify any groups of functions that are Θ of one another.

n	2^n	n^3	$n\ln(n)$	2	$n!$	$\log_2((4n)^n)$	$\ln(n^2)$	$\left(\frac{3}{2}\right)^n$	$n^{1/5}$	$\ln^2(n)$	52!

2. Consider the following intuition for a sorting algorithm. Let A be a list of real numbers. If A is of size 0 or 1, return it since it is already sorted. Otherwise, pick the last element of A to be used as a pivot and call it p. For each element e of A except the last element, if $e \leq p$, place e in a list called L. On the other hand, if $e > p$, place e in another list called R. Repeat this procedure on L and R and call the resulting lists L' and R'. Make a new list by adding p between L' and R'. Return the result.

   (a) (10 pts) Write pseudocode following the above intuition.

   (b) (5 pts) Determine the **worst-case** asymptotic run time of this algorithm and explain why this is the worst case.

   (c) (5 pts) Assume that the sizes of L and R are equal at each step of the algorithm. Find a recurrence relation describing the run time in this case.

   (d) (10 pts) Given this recurrence relation, what is the asymptotic run time of the algorithm? Be sure to justify your answer using the master theorem.

3. The computer science department keeps a running record of Mario Kart match-ups and who has beaten who. In particular, there is a graph $G = (V, E)$ where each player is a vertex and $(u, v) \in E$ for $u, v \in V$ when u has beaten v at least as many times as v has beaten u. Based on this information, it is not always possible to determine a single best player (Why not? Under what circumstances is this possible?). However, we can split players into three groups, A, B, and C, as follows. We say that player u weakly dominates player v, denoted $u \rightsquigarrow v$, when there is a path from u to v in G.

   - A is the set of players u such that (1) if $v \rightsquigarrow u$, then $u \rightsquigarrow v$ and (2) there is some player w such that $u \rightsquigarrow w$ but $w \not\rightsquigarrow u$.
   - B is the set of players u such that (1) if $u \rightsquigarrow v$, then $v \rightsquigarrow u$ and (2) there is some player w such that $w \rightsquigarrow u$ but $u \not\rightsquigarrow w$.
   - C is the set of all players not included in A or B.

   There are several specific examples of A, B, and C at the end of this document.

   (a) (5 pts) Describe what the sets A, B, and C mean in the context of Mario Kart along with an intuitive approach for determining the size of these sets.

   (b) (10 pts) Write pseudocode for a function `getSetSizes(G)` which returns $(|A|, |B|, |C|)$, a triple with the sizes of each set. You may assume the existence of a `makeSCCs(G)` function which returns the SCC metagraph of G.

   (c) (5 pts) Analyze the asymptotic run time of your algorithm.

   (d) (10 pts) Write pseudocode for a function `rankPlayers(G)` which returns a list of nodes such that all nodes in group A appear at the beginning, all nodes in group B appear at the end, and, for all other nodes, u appears before v if $u \rightsquigarrow v$ and $v \not\rightsquigarrow u$. Intuitively, this order is one possible ranking of players. As in (b), you may assume the existence of a `makeSCCs(G)` function.

Coding Problem

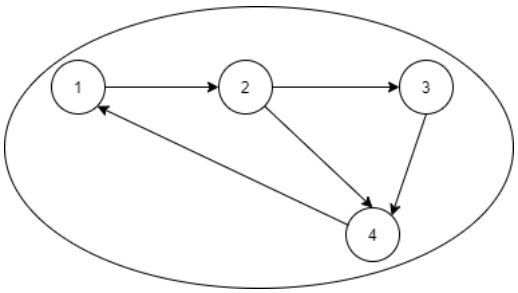
(20 pts) **Mario Kart Rankings:** Write a C++ implementation of the pseudocode you developed for problem (3b) and submit to INGInious as assignment_1.cpp. Some skeleton code that you might find useful is available on Canvas (assignment_1_skeleton.cpp).

- Input will come from cin
  - The first line will contain two integers, n and m, separated by a space.
  - n is a number of vertices and m is a number of edges.
  - The next m lines will contain two integers, u and v, separated by a space.
  - Each of these pairs represents a directed edge (u, v).
- Print output to cout
  - On one line print the sizes of the three sets following the format "$|A|$ = `A size`, $|B|$ = `B size`, $|C|$ = `C size`" with no trailing newline.

Examples

In the following examples, green nodes belong to A, red nodes belong to B, and white nodes belong to C.

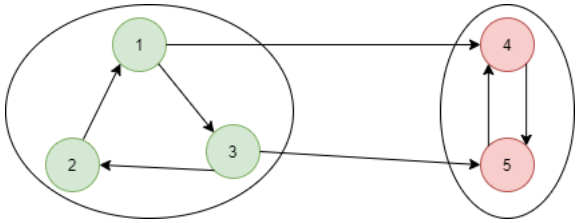
Example 1:



Input:
4 5
1 2
2 3
3 4
4 1
2 4

Expected output:
$|A| = 0$, $|B| = 0$, $|C| = 4$

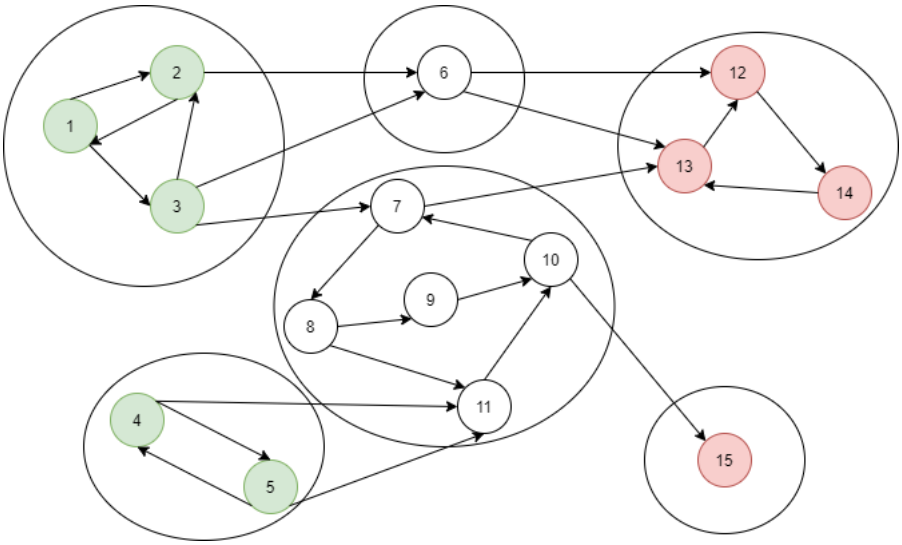
Example 2:



Input:
5 7
1 3
3 2
2 1
1 4
3 5
4 5
5 4

Expected output:
$|A| = 3$, $|B| = 2$, $|C| = 0$

Example 3:



Input:
15 24
1 2
2 1
1 3
3 2
3 6
2 6
6 12
6 13
13 12
12 14
14 13
3 7
7 13
7 8
8 9
8 11
9 10
11 10
10 7
4 5
5 4
4 11
5 11
10 15

Expected output:
$|A| = 5$, $|B| = 4$, $|C| = 6$