<u>IDC – Algorithms – Homework 1</u>

Due date: Sunday 13/11/2022

Guidelines:

- You may answer any problem or subproblem with “I don’t know” and get half the points (rounded down). The total number of points you can accumulate this way is 25 points per homework assignment.
- Do not submit a solution consisting of only “I don’t know”s. When we calculate final grades, we will regard any unsubmitted assignment as 25.

Collaboration policy:

- Each student should submit an individual solution.
- You may discuss homework problems in broad terms with other students.
- You may not discuss every last detail, dictate or share written solutions. Do not write your solution together with other students.
- We are going to be strict here. The goal of the homework is to give you an opportunity to struggle with solving problems and formulating your ideas by yourself.

Technical requirements:

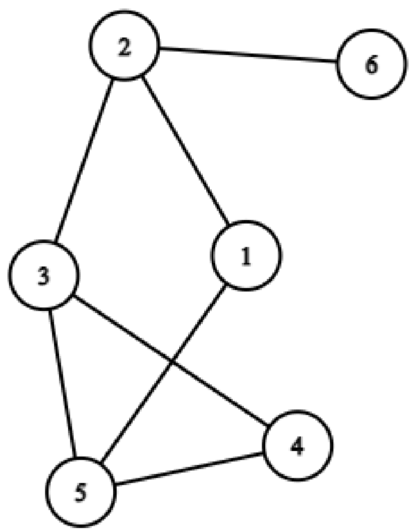
- You must fill in your answers on this form in the allocated spaces (or, if you use LaTeX, copy-paste the statements of the problems; no need to fix corrupted formulas). You may add extra space if you need, but too long answers will result in point deduction.
- Submit your solution as a single PDF file of size up to 5MB. The grading app doesn’t support other formats and doesn’t cope well with large files. Google “shrink PDF file size” if needed.
- The page size of your PDF file should be standard A4 and the orientation should be portrait. Before submitting your PDF, open it with Adobe Acrobat Reader, go to File > Properties > Description, and make sure Page Size is around 21 × 29 cm (again, the grading app doesn’t support zooming). Please note that scanning A4 pages doesn’t guarantee that the resulted PDF file has an A4 page size. Google “resize PDF to A4” if you need.
- Make sure pages are properly ordered and oriented, and text is highly readable: large enough, sharp, in focus, and in high contrast with the background.
- Do not squeeze scanned pages to fit in the allocated spaces; the text becomes too small.
- Do not type in your answers as PDF comments (the grading tool ignores comments).
- Submit your solution electronically via the course Moodle page. After submitting you should download the submitted file back from Moodle to make sure it is the right file.
- Failing to meet any of the technical requirements will result in a fine or your submission not being graded at all.
- For clarifications or any other help, you are welcome to use the course forum in Piazza.

Problem 0 (4 pts):

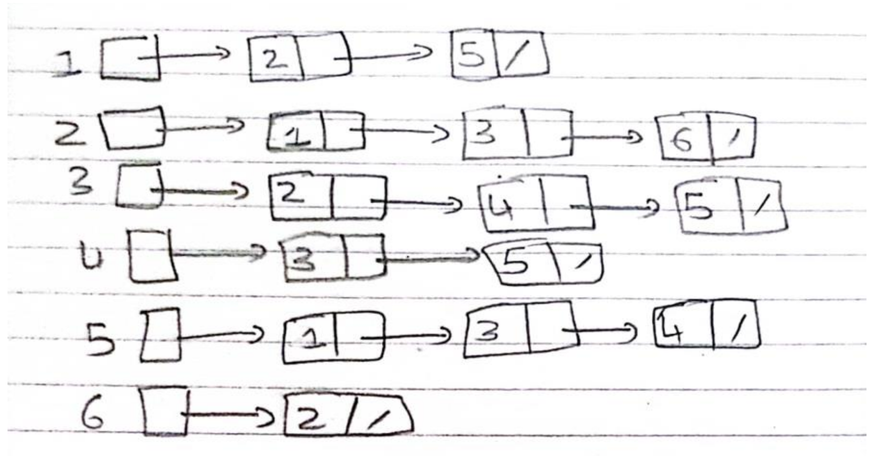
1. Have you read the instructions on page 1 carefully? Yes
2. What app allows you to check the Page Size of your PDF? Adobe Acrobat Reader
3. What should be the page width (**in centimeters**) of your submitted file? 21cm
4. What might happen in case you fail to meet any of the technical requirements? I will get a fine or my submission will not be graded
5. How many points (more or less) will you lose from the final grade in this course, if you answer "I don't know" on one question worth 25 points in each of the HW assignments in this course (recall that the HW grade determines 15% of the final grade)? 13.13%

Problem 1 (22 pts):

An undirected graph G is given in the figure.



a. Draw the representation of G by adjacency lists. What is the size of the data structure if it takes 24 bits to store a header node (including a pointer to its adjacency list), and 32 bits to store a cell in a list? (use the actual number of vertices and edges in G to give your answer).



Size: (24) |V|+ 2 (32)|E| = (24) (6)+ 2 (32)(7) = 592 bits

Convert G into a directed graph G': Let M be the adjacency matrix of G, then the adjacency matrix M' of G' is defined as follows: M'[i,j]=1 if and only if M[i,j]=1 and i<j.

i. Repeat part ***a*** of the problem for G'.

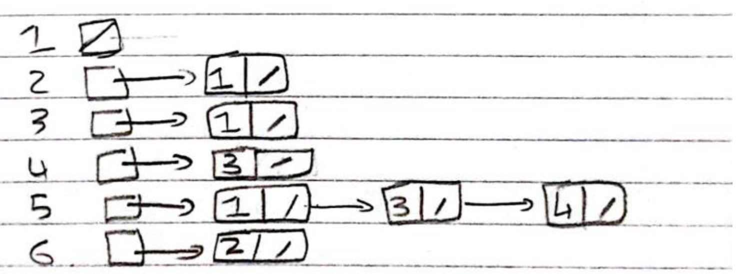
M:

	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	0	0	1	0	1	0
5	1	0	1	1	0	0
6	0	1	0	0	0	0

M':

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	0	1	0	0	0	0
4	0	0	1	0	0	0
5	1	0	1	1	0	0
6	0	1	0	0	0	0

G':



Size: (24) |V|+ (32)|E| = (24) (6)+ (32)(7) = 368 bits

ii. Write all the topological sorts of G'

iii. Write all the edges for which the following holds: If the direction of this edge (only) is reversed then G' has no topological sort.

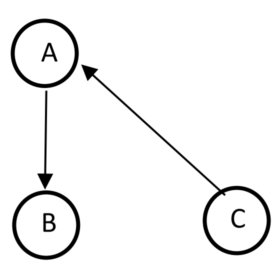
ii. 6,2,5,4,3,1 6,5,2,4,3,1 6,5,4,2,3,1 6,5,4,3,2,1 5,6,2,4,3,1 5,6,4,2,3,1 5,6,4,3,2,1 5,4,6,2,3,1 5,4,6,3,2,1 5,4,3,6,2,1

iii. (3,5) , (5,1)

b. Prove or give a counter example: If G_1 is a DAG with only one topological sort, any addition of an edge that keeps the graph simple (no parallel edges and no self-loops) results in a graph without a topological sort.

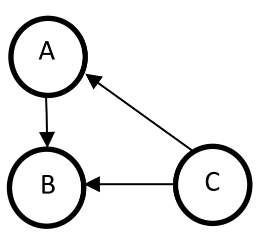
False, counterexample:

G_1 :



Has only one Top. sort of C,A,B

G_1 with an added edge:



Still has a Top. Sort of C,A,B

Problem 2 (21 pts):

SpongeBob and Patrick need to find topological sorts of a k different and disjoint DAGs, $G_1 = (V_1, E_1), \dots, G_k = (V_k, E_k)$. They decide to use the algorithm we saw in class. SpongeBob runs the algorithm separately for each graph. Patrick allocates and initializes one in-Degree array D of size $\sum_i |V_i|$ and performs only one run of the algorithm. Whenever a vertex is output in Patrick's run, it is added to the output of its corresponding graph.

1. Does the algorithm of Patrick return a correct output? Prove or give a counter example.
2. Analyze the time complexity of SpongeBob's method, and of Patrick's method (as a function of $|V_1|, |V_2|, \dots, |V_k|, |E_1|, |E_2|, \dots, |E_k|$).

1. I don't know

2. SpongeBob's: O(max(| V_1 | + | E_1 | ,| V_2 | + | E_2 | , … ,| V_k | + | E_k |))

For any graph, it takes O(|Vi| + |Ei|) to initialize the In-Degree array and all the other processes done in the algorithm is less than O(|Vi| + |Ei|). Therefore the runtime for any single graph is O(|Vi| + |Ei|). This method is performed k times (for the k graphs), therefore the run time is <= k(max(| V_1 | + | E_1 | ,| V_2 | + | E_2 | , … ,| V_k | + | E_k |)), therefore the runtime is O(max(| V_1 | + | E_1 | ,| V_2 | + | E_2 | , … ,| V_k | + | E_k |)).

Patrick's: O(max(| V_1 | + | E_1 | ,| V_2 | + | E_2 | , … ,| V_k | + | E_k |))

To initialize the whole D array, it takes O($\sum_i (|V_i| + |E_i|)$)= O(k (max($|V_1| + |E_1|$,…, $|V_k| + |E_k|$))). All the other process take less than O(|Vi| + |Ei|). Therefore, for the same reasoning as in SpongeBob's runtime, the answer is O(max($|V_1| + |E_1|$,…, $|V_k| + |E_k|$)).

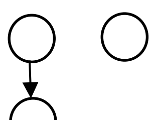
Problem 3 (3*7=21 pts):

For each of the following statements, determine if it is true or false. Justify by proving a true statement or providing a counter example for a false statement.

Definition: A "super-source" in a directed graph is a vertex v for which out-deg(v)=n-1 and in-deg(v)=0.

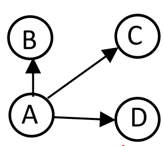
1. Every DAG has at least one "super-source".
2. Every DAG has at most one "super-source".
3. The topological sort of a DAG that has a single super-source is unique, independent on the data structure used by the algorithm.

1. False, counterexample:



2. True: ATC that there is 2 super-sources, A and B. Since out-deg(A)=n-1, there is an edge going from A to every other vertex in the graph, and therefore has to have an edge pointing to vertex B. Therefore in-deg(B) >=1. Contradiction to B being a super-source (i.e having in-deg(B)= 0).

3. False, Counterexample:



Node A is the single super source in this DAG, however there is more than 1 topological Sort. Sort 1 is A,B,C,D and Sort 2 is A,C,B,D etc.

Problem 4 (18 points(3*6):

a. Let G=(V,E) be an undirected graph and let $A, B \subseteq V$ be disjoint subsets of V such that for every $e \in E$, $|e \cap A| = |e \cap B| = 1$. Prove $\sum_{a \in A} \deg(a) = \sum_{b \in B} \deg(b)$

For every e ∈ E, +1 is added to both vertices that is incident to the edge. Since $|e \cap A| = |e \cap B| = 1$, and that A and B are disjoint, we know that 1 of the vertices is in A and the other is in B. Therefore $\sum_{a \in A} \deg(a) = \sum_{b \in B} \deg(b)$ as both the degree of A and B is being incremented for every e ∈ E.

b. For each of the following descriptions of graphs, give an example of such a graph, or prove one cannot exist.

1. An undirected graph with 5 vertices, each with a degree of 3.

This graph cannot exist. Proof:

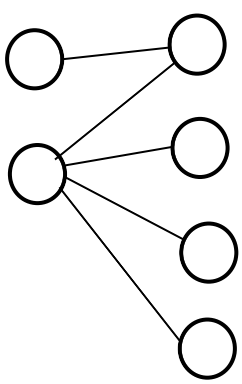
By the handshaking theorem $2|E| = \sum_{v \in V} \deg(v)$

Therefore 2|E| = 3(5)

2|E| = 15

|E| = 7.5 which cannot exist

2. a connected bipartite graph with 6 vertices, where exactly one vertex has a degree of 4.

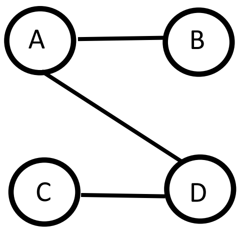


Problem 5 (18 points):

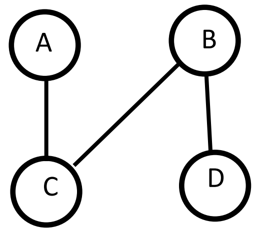
Let G = (V, E) be a simple undirected graph. The **complement graph** of G is defined as follows: $G^c = (V, E^c)$, where $(u,v) \in E^c$ iff $(u,v) \notin E$. That is, the set of nodes of G^c is the same as that of G, and there is in an edge in G^c connecting any two nodes that do not have an edge connecting them in G.

a. (6 pts.) Describe an undirected graph G with n=4 nodes, such that both G and G^c are connected. Draw both G and G^c.

G:



G^c:



b. (12 pts.) Prove: For every simple graph with n>2 nodes, if G^c is not connected, then G is connected.

We will prove by induction:

Base Case: n=3

When theres only 3 nodes, G^c can only have one edge in order for it to be disconnected, as any more edges would result in a connected graph. Therefore the compliment graph G would contain the other two possible edges in a 3 node graph, and therefore is connected as a 3 node graph with 2 edges has to be connected.

Induction Hypothesis: Let B^c be a disconnected graph with n+1 nodes, we must show that its compliment graph B is connected

Induction Step: Let G^c be a subgraph of B^c that contains n nodes and let the excluded node be V1

Scenario 1: if G^c is a connected graph

Then that means that the V1 does not have any edge connecting it to G^c, because otherwise B^c would be connected (adding an edge from any node to a connected graph, is also a connected graph). Therefore in the compliment graph B, V1 has edges going from V1 to every other node and therefore B is connected.

Scenario 2: if G^c is a disconnected graph

Then its compliment graph G is connnected. Note that in this scenario, V1 doesn't have every edge possible for it, as if it did, B^c would be connected. Therefore in the compliment graph B, V1 has atleat one edge joining it to the compliment subgraph G, making the whole of B connected (as a node with an edge to a connected graph, is also a connected graph)