# COL 351: Analysis and Design of Algorithms
# Minor Exam

Timing: 9:10 am - 10:40 am

**Important Guidelines:**

1. This is a closed book exam. You cannot look at your notes or browse the internet.
2. Each solution must start on a new page.
3. You can directly reference an algorithm / theorem proved in the lectures.
4. Total marks are 42.
5. The scanned solution **must be** uploaded on Gradescope by 10:55 am sharp. The late submission deadline is 11:00 am, No submissions will be accepted later in any circumstance.

## 1 Short Questions $[3 \times 5 = 15$ marks$]$

(a) Prove that edit-distance between two strings $A,\ B$ of size respectively $m, n$ is always bounded by $(m + n - 2|LCS(A, B)|)$, where $|LCS(A, B)|$ is the length of longest common subsequence of $A$ and $B$ (when these strings are viewed as sequences).

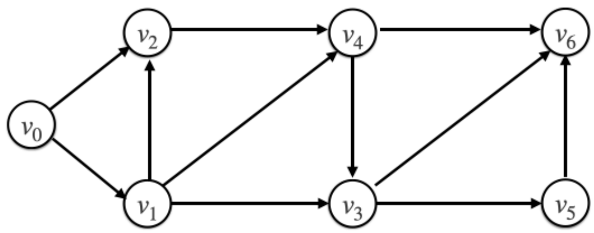
(b) An edge $(x, y)$ in an undirected graph $G$ is said to be a *bridge* if each path from $x$ to $y$ in $G$ passes through edge $(x, y)$.

Prove that an edge is a bridge if and only if it does not lie on any simple cycle of $G$.

(c) Device an $O(m+n)$ time algorithm that verifies whether a given directed graph with $n$ vertices and $m$ edges is strongly connected (i.e. has exactly one SCC) using only BFS traversal.(Also provide short correctness argument).

(d) Device the most efficient algorithm to verify whether a given directed graph with $n$ vertices and $m$ edges has a unique topological ordering. (Also provide a very short correctness proof of your algorithm).

(e) Draw a DFS tree of the graph below with respect to source $v_0$. Also provide start-time and finish-time of all the vertices with respect to your DFS tree. Compute a topological ordering of the graph using the finish-time of vertices with respect to the DFS tree compute by you.




## 2 Gemstones [6 marks]

You are given set of $k$ precious gemstones and a function $F : \mathbb{N} \to \mathbb{N}$ where $F(i)$ denotes the market price of a box of $i$ gemstones (for $1 \leq i \leq k$). Your task is to find the optimal way to partition $k$ gemstones into smaller sets that on selling provides you maximum overall profit.

Design an $O(k^2)$ time algorithm to output the optimal partitioning for $k$ gemstones in a 'list'. Also provide a short correctness argument.

## 3 Distances $[(3 + 3 + 2 + 4) = 12$ marks]

$G = (V, E, wt)$ is a directed, strongly-connected, weighted graph with $n$ vertices and $m$ edges, where $wt$ is the edge-weight function. It is given that $G$ has no cycle of negative weight. Let $s \in V$ be a vertex in $G$. Define a **new** weight function $wt^*$ as follows:

$$wt^*(x, y) := wt(x, y) + dist_G(s, x) - dist_G(s, y).$$

(a) Show that for each edge $(x, y) \in E$, $wt^*(x, y) \geq 0$. Further, the values $wt^*(e)$, for $e \in E$, are computable in $O(mn)$ total time.

(b) For a path $P$, define $wt(P) := \sum_{e \in P} wt(e)$ and $wt^*(P) := \sum_{e \in P} wt^*(e)$.

Prove that for any $P = (v_0, \ldots, v_t)$, $wt^*(P) = wt(P) + dist_G(s, v_0) - dist_G(s, v_t)$.

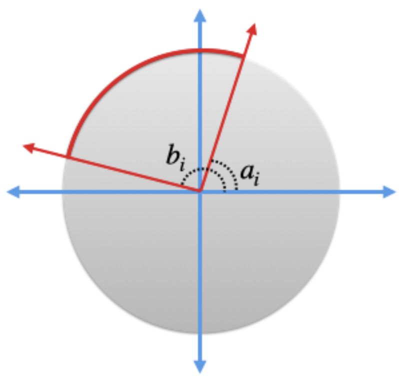
(c) Use (b), to argue that for $x, y \in V$, a path $Q$ is a shortest path from $x$ to $y$ according to weight function $wt$ if and only if $Q$ is a shortest path from $x$ to $y$ according to weight function $wt^*$.

(d) Present an $O(mn \log n)$ time algorithm to compute the distance between all the vertex-pairs in the input graph $G$. Also provide correctness argument.

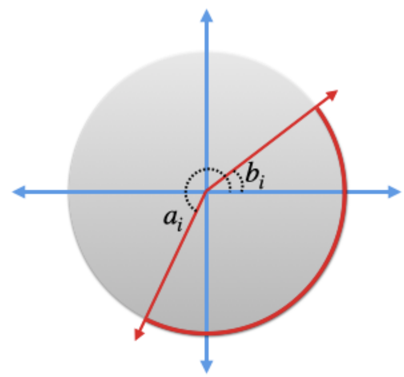
# 4 Covering the circumference [$6 + 3 = 9$ marks]

Let $C$ be a unit circle on $x - y$ plane centered at the origin. You are given $n$ arcs on $C$ numbered $1, \ldots, n$. For $i \in [1, n]$, the arc $i$ starts at an angle $a_i$ from $x$-axis (in anticlockwise direction) and goes up to angle $b_i$ (in anticlockwise direction).

Further, it is given to you that the union of all the arcs covers the entire circumference of the circle.




$a_i < b_i$




$a_i > b_i$

(a) Device an $O(n \log n)$ time algorithm that given an arc $(a_{i_0}, b_{i_0})$, finds a subset $S$ of the minimum possible size such that $S \cup \{ (a_{i_0}, b_{i_0}) \}$ covers the circumference of the circle. (Also provide a correctness argument).

(b) Use (a) to device an $O(n \log n)$ time algorithm to find a subset $R$ of arcs satisfying:

- union of arcs in $R$ cover the circumference of the circle, and
- $|R|$ is at most one larger than the minimum number of arcs needed to cover the circumference of the circle.

(Also provide a short correctness argument).