School of Computer Science

Carleton University

Name: **A.M.** Problem Set: **4**
Course: **COMP 3804** Week: **1-4**

Problems

Consult [1, 2, 3].

Problem 1 (Section 4.1 of [1]). *Let A be an array of size n, where each $A[i]$ is an integer (positive or negative) and $1 \leq i \leq n$. For $1 \leq i \leq j \leq n$, define $\Delta(i,j) = \sum_{k=i}^{j} A[k]$. Find a maximum subarray of A, i.e., find a pair of indices α and β, where $1 \leq \alpha \leq \beta \leq n$, such that $\Delta(\alpha,\beta) \geq \Delta(i,j)$, for all possible choices of $1 \leq i \leq j \leq n$. Devise first a naive algorithm running in $O(n^3)$ time by considering all possible choices of i and j. Devise an $O(n \log n)$ time divide-and-conquer algorithm. Can you come up with an algorithm running in $O(n)$ time? See Exercise 4.1-5 of [1].*

Problem 2. *Assume that you have a set of three parallel and distinct lines in plane. Assume that Line 1 contains a set R of n red points, Line 2 contains a set B of n blue points, and Line 3 contains a set G of n green points, where n is a positive integer. You need to design an algorithm that determines if there is a triplet of points (r,b,g) such that they lie on a line segment, where $r \in R$, $b \in B$ and $g \in G$. (Observe that it is straightforward to design an algorithm running in $O(n^3)$ time by considering each choice for points r, b, and g, and in $O(1)$ time testing whether they all lie on a segment.) To get Bonus points, design an algorithm running in $O(n^2)$ time.*

Problem 3. *This problem is related to the representation of graphs. Assume that the number of edges in the graph $G=(V,E)$ is small, i.e., it is a sparse graph. In the adjacency matrix representation of G, the normal tendency is to first initialize the matrix, requiring $O(|V|^2)$ time. Is there any way we can initialize the adjacency matrix in time proportional to $O(|E|)$ and still have $O(1)$ adjacency test?*

Problem 4. *Show that in a depth-first search, if we output a left parenthesis '(' when a node is accessed for the first time and output a right parenthesis ')' when a node is accessed for the last time, then resulting parenthesization (or bracketing sequence) is proper. Each left '(' is properly matched with a right ')'.*

Problem 5. *Consider the following modified pseudo-code.*

Modified DFS Algorithm
Input: A graph $G=(V,E)$, represented by adjacency list $L[v]$ for each vertex $v \in V$.
Output: A pair of integers $(pre[v], post[v])$ assigned to each vertex v.

1. *Clock* $:= 1$;
2. *for all $v \in V$ do mark v as* unvisited;
3. *While there exists an unvisited vertex v do SEARCH(v)*

procedure SEARCH(v)

1. mark v as *visited*;
2. $pre[v] := Clock$;
3. $Clock := Clock + 1$;
4. for each vertex w on $L[v]$ do
   if w is unvisited then SEARCH(w);
5. $post[v] := Clock$;
6. $Clock := Clock + 1$;

Answer the following questions:

1. *Suppose $G = (V, E)$ is undirected graph. Show that for any pair of nodes u and v in G, the two intervals $[pre[u], post[u]]$ and $[pre[v], post[v]]$ are either disjoint or one interval contains the other.*
2. *Execute the modified dfs algorithm on a few directed graphs.*
3. *Suppose $G = (V, E)$ is directed graph. Show that for any pair of nodes u and v in G, the two intervals $[pre[u], post[u]]$ and $[pre[v], post[v]]$ are either disjoint or one interval contains the other. Moreover, show that if for a directed edge $(u, v) \in E$, $pre[v] < pre[u] < post[u] < post[v]$, then there is a directed cycle in G.*
4. *Call an edge $e = (u, v)$ of a directed graph a* back edge *if $pre[v] < pre[u] < post[u] < post[v]$. Show that a directed graph has a directed cycle if and only if the modified dfs algorithm reveals a back edge.*
5. *Design an algorithm that determines whether a directed graph $G = (V, E)$ is an acyclic graph (i.e., it doesn't contain a directed cycle). Your algorithm must run in $O(|V| + |E|)$ time.*
6. *Let $G = (V, E)$ be a directed acyclic graph. Show that for any directed edge $e = (u, v) \in E$, $post[u] > post[v]$.*
7. *All vertices with no incoming edges in a directed acyclic graphs are called the* source *vertices, and all the vertices that have no outgoing edges are called the* sink *vertices. In any directed acyclic graph, can you say what property the vertex with the largest post number satisfies? the vertex with the smallest post number? Does the ordering of the vertices with respect to decreasing post number results in a linear order?*

Problem 6. *Let $G = (V, E)$ be a directed graph given in the adjacent matrix representation. Define the square of G to be the graph $G' = (V', E')$ where $V' = V$ and $(u, v) \in E'$ if and only if there is a directed path consisting at most two edges between u and v in G. Given G show how you can compute G' efficiently.*

References

[1] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms.* 3rd. ed., MIT Press, 2009.

[2] S. DasGupta, C. Papadimitriou, V. Vazirani. *Introduction to Algorithms.* McGraw Hill.

[3] A. Maheshwari. *Notes on Algorithm Design*, Chapter 1, `https://people.scs.carleton.ca/~maheshwa/Notes/DAA/notes.pdf`