Homework 1: Due 11:59pm Friday, February 9, 2024

(Problem 5 is required for graduate students. Undergraduate students who solves Problem 5 can earn at most 5 points extra credit.)

Problem 1: (30 points)
For each pair of A and B in the list below, indicate their asymptotic relation (O, Ω, Θ). No justification is needed. (Assume that $k \geq 1, c > 1$ are constants.)

1. $A = n^{10}, B = n(\log n)^{100}$;
2. $A = n^k, B = c^n$;
3. $A = n^{0.1}, B = 2^{((\log_2 n)^{10})}$;
4. $A = \sqrt{n}, B = n^{\sin n}$;
5. $A = \lg(n!), B = \lg(n^n)$.

(Hint: It is possible that none or more than one of O, Ω and Θ relations hold for a pair of A and B.)

Problem 2: (20 points)
You are given a list of integers $a_1, a_2, \ldots, a_n$. You need to output an $n \times n$ matrix A in which the entries $A[i,j] = a_i + a_{i+1} + \cdots + a_j$ for $i < j$ (for $i \geq j$, $A[i,j]$ is 0). Consider the following algorithm for this problem.

Set $A[i,j] = 0$ for all $1 \leq i, j \leq n$
for $i = 1$ to n **do**
  for $j = i + 1$ to n **do**
    for $k = i$ to j **do**
      $A[i,j] = A[i,j] + a_k$
    end for
  end for
end for

---

1. What is the worst-case running time of above algorithm? Give the running time as a function of n.
2. Design an algorithm for this problem with asymptotically faster running time than above algorithm. What is the running time of your new algorithm?

Problem 3: (20 points)
Give an algorithm to detect whether a given undirected connected graph contains a cycle. If the graph contains a cycle, then your algorithm should output **one cycle** (do not output all cycles in the graph, just any one of them). Justify the running time bound of your algorithm.
You will receive 20 points if the running time of your algorithm is $O(m + n)$ for a graph with n nodes and m edges. You will receive 15 points if your solution is correct, and the running time of your algorithm is polynomial in n and m.
(*Hint: Graph traversal.*)

Problem 4: (20 points)
Given a **directed** graph $G = (V, E)$ with **no cycle**. Give an algorithm to check if there is a **directed** path that touches every vertex **exactly once**.
(*Hint: Topological sort.*)

Problem 5*: (25 points for graduate students, 5 extra points for undergraduate students)
Given a connected graph G with n vertices. We say an edge of G is a **bridge** if the graph becomes a disconnected graph after removing the edge. Give an algorithm that finds all the bridges. Justify the running time bound of your algorithm.
You will receive 25 points if the running time of your algorithm is $O(m + n)$ for a graph with n nodes and m edges. You will receive 18 points if your solution is correct, and the running time of your algorithm is polynomial in n and m.
(*Hint: DFS.*)