Homework 1 Solution

Problem 1

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

Solution:

1. $A = O(B)$;
2. $A = O(B)$;
3. None;
4. $A = O(B), A = \Omega(B)$ and $A = \Theta(B)$.

Problem 2

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**
            $A[i,j] = A[i,j] + a_k$

---

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?

Solution:

1. $O(n^3)$.
2. $O(n^2)$.

Set $A[i,j] = 0$ for all $1 \le i, j \le n$
for $i = 1$ to n **do**
    for $j = i + 1$ to n **do**
        $A[i,j] = A[i,j-1] + a_j$

Problem 3

The following statements are correct or not. If yes, explain why. If not, give a counterexample.

1. If an n vertices graph G has at least n edges, then it has a cycle.
2. In any tree with n vertices, the number of nodes with degree 8 or more is at most $(n-1)/4$.

Solution:

1. True. If the graph is not a connected graph, then we choose a connected component in which the number of edges is greater than or equal to the number of vertices. Let k denote the number of vertices in the connected component. The number of edges in the connected component is at least k. Since a connected graph with no cycle is a tree, then if the connected component does not have any cycle, the connected component has $k-1$ edges. This contradicts to the fact that the connected component has at least k edges.

   (If the student only consider the case that G is a connected graph, also give the full score.)

2. True. Since a tree with n vertices contains $n-1$ edges, we have

$$\sum_v \deg(v) = 2(n-1).$$

   On the other hand,

$$8 \cdot |\{v : \deg(v) \ge 8\}| = \sum_{v:\deg(v)\ge 8} 8 \le \sum_v \deg(v).$$

   So we have

$$|\{v : \deg(v) \ge 8\}| \le 2(n-1).$$

Problem 4

Give an algorithm to detect whether a given undirected graph contains a cycle. If the graph contains a cycle, then your algorithm should output one cycle (it need not output all cycles in the graph, just one of them). The running time of your algorithm should be $O(m+n)$ for a graph with n nodes and m edges.

Solution: (There are different correct solutions, e.g. using BFS. Please check the solutions from students carefully.)

An $O(m+n)$ running time algorithm:

1. Run DFS on the graph, and let T denote the DFS tree.
2. If $G = T$ (all the edges of G are in T), then return No.
3. Otherwise, let (u, v) be an arbitrary edge of G that is not in T.

4. Find the path from v to u in T, let P denote this path.
5. Return $P \cup \{(v,u)\}$.

Problem 5

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 $O(m+n)$ time algorithm that finds all the bridges. (Partial credits will be given for a polynomial time algorithm.)
(*Hint: Use DFS.*)

Solution: (Please give 18 points for students who give a correct algorithm, but the running time is slow.)
An $O(nm)$ running time algorithm:

1. Enumerate every edge (u,v) in the tree
   (a) Delete edge (u,v) in the graph.
   (b) Run BFS with start vertex u.
   (c) If the BFS visited vertex v, then (u,v) is a bridge.
   (d) Add edge (u,v) back to the graph.

An $O(n+m)$ running time Algorithm:

1. Run DFS with an arbitrary vertex s as the root. For every vertex v, set $number(v)$ to be integer i if v is the i-th discovered vertex in the DFS.
2. For $i = n, n-1, \ldots, 1$
   (a) Let v be the vertex such that $number(v) = i$ and set $back(v) = i$
   (b) For each edge (v,u) which is a DFS tree edge, such that u is a child of v, set
   $$back(v) \leftarrow min\{back(v), back(u)\}$$
   (c) For each edge (v,u) which is not a DFS tree edge, set
   $$back(v) \leftarrow min\{back(v), number(u)\}$$
3. For each vertex v
   (a) If $v \neq s$ and $back(v) = number(v)$, then (v,u) is a bridge, where u is the parent of v.