Informatics 2 – Introduction to Algorithms and Data Structures

Tutorial 5: Sorting and Graph Algorithms

This sheet covers material from Lectures 13, 14 and 15.
The question with $^{(*)}$ might be more challenging.

1. Assume that you are given an array A containing n integer numbers from the set $\{0, 1, \ldots, k\}$ for some $k \leq n$. Search on the internet or the textbooks for an algorithm that on input A produces a sorted array containing those n numbers in non-decreasing order in time $O(n)$. Present the pseudocode for the algorithm and provide a justification for its running time.

   This algorithm sorts the array in $O(n)$ time. Does this contradict the lower bound of $\Omega(n \lg n)$ on the running time of any general sorting algorithm presented in the lectures?

2. Consider the graph G of Figure 1.

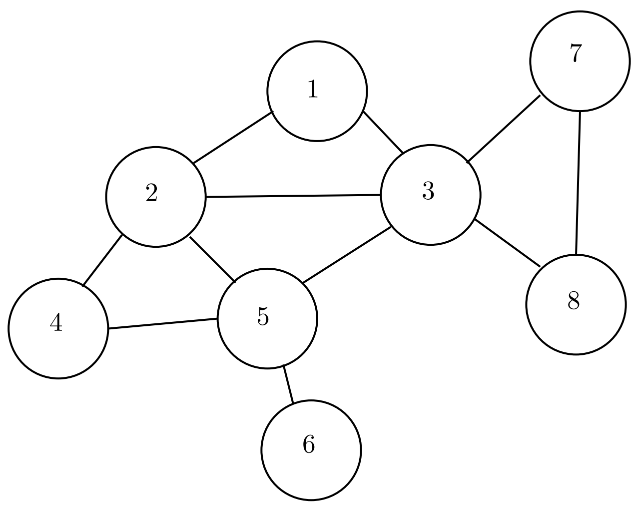

Figure 1: The graph G (part of the graph in Fig 3.2. in KT)

   (a) Write the adjacency matrix and the adjacency list representation for the graph. For the adjecency list, for consistency, consider the neighbours of a vertex appearing in ascending numerical order in the list for each node.

(b) Run Depth-First Search (DFS) on G starting from node 1. Explain the steps of DFS and note the order in which the nodes will be explored by DFS. Write down the spanning tree produced by DFS in adjacency list representation.

(c) Run Breadth-First Search (BFS) on G starting from node 1. Explain the steps of BFS and note the level that each node will be assigned to during the execution of BFS. Write down the spanning tree produced by BFS in adjacency list representation.

(d) Compare the two spanning trees produced by DFS and BFS starting from node 1 respectively. What do you observe?

3. (a) Prove the following property for the layers produced by BFS: For any edge (u, v), either u and v are in the same layer, or $|L(u) - L(v)| = 1$, where $L(x)$ is the layer of node x.

   (b) $^{(*)}$ Let $G = (V, E)$ be a connected graph and let $s \in V$ be a node of G. Suppose that we run DFS(G, s) and obtain a DFS spanning tree T and that we also run BFS(G, s) and obtain the same BFS spanning tree T. Prove that $G = T$.

4. (a) Assume that you are given access to a graph $G = (V, E)$ and a node $u \in V$, and for every node $v \in V$, the distances $d(u, v)$ between u and v in G. Present an algorithm that produces a sorted list of the nodes in terms of non-decreasing distance from u which runs in time $O(|V|)$. Recall that the distance between two nodes is the number of edges in the shortest path between the nodes in G.

   (b) Assume that you are given access a the graph $G = (V, E)$ and a node $u \in V$ as above, but not the distances $d(u, v)$ as above. Present an algorithm that produces a sorted list of the nodes in terms of non-decreasing distance from u which runs in time $O(|V| + |E|)$.