Data Structures and Algorithms

Practice-sheet

Breadth First Search Traversal

Note: Each graph is represented as adjacency list unless mentioned otherwise. Furthermore, n denotes the number of vertices and m denotes the number of edges. All the problems in the following section just require a good understanding of the BFS traversal. So there is absolutely no need of hint or model solution. However, if after spending sufficient time you are not able to solve a problem, you may discuss it in the doubt clearing session to be held on 15th October.

Breadth First Search (BFS) Traversal

1. **Running time of BFS traversal**
   What is the running time of the BFS traversal algorithm if the underlying graph is represented by an adjacency matrix ?

2. **BFS traversal of directed graph**
   We discussed BFS traversal from a vertex in an undirected graph. The same algorithm works for directed graphs as well. What relationship can you establish on the levels of the endpoints of an edge ?

3. **Critical edges**
   You are given an undirected connected graph $G = (V, E)$ and vertex $s \in V$. A single edge deletion may lead to increase in the distance from s to one or more vertices in the graph. Our aim is to find out all such edges in the graph G. Design an $O(m + n)$ time algorithm to compute all such edges.

4. **Detecting cycle in a graph**
   Design an $O(n)$ time algorithm to determine if a given undirected graph on n vertices has a cycle.

5. **Extension of the algorithm to determine if a graph is a bipartite graph**
   A graph is said to be connected if there is a single connected component in the graph. In the class, we designed an $O(m+n)$ time algorithm to determine if a given connected graph is bipartite. How can you extend the algorithm to handle the case when the given graph is not connected (there are two or more connected components in the graph) ?

6. **Extension of BFS for weighted graphs**
   Suppose there is a connected graph where each edge has length which is a positive integer less than c. So we can define length of a path as the sum of the length of all edges lying on the path. Design an $O(mc + n)$ time algorithm to compute distance (length of the shortest path) from a given fixed vertex x to all other vertices in the graph. Your algorithm may use $O(mc + n)$ extra space if you wish.

7. **A surprising algorithm**

   You are given a graph $G = (V, E)$ and two vertices u and v. Furthermore, you are told that u and v belong to different connected components. The size of a connected component is defined as the number of edges present in that component. You have to determine whether the component of u is smaller than the component of v. Design a simple algorithm for this problem whose running time is of the order of the size of the smaller component. The algorithm should be easily implementable in practice.

8. **A data structure for the shortest paths**
   Design an $O(mn)$ time algorithm which computes an $O(n^2)$ space data structure for a given graph $G = (V, E)$ such that given any two vertices $u, v \in V$, the data structure can output the shortest path between u and v. The time taken by the data structure to output the path has to be of the order of the number of vertices on the path.

9. **A prime path**
   (*This problem will make you realize that many problems can be modeled as graph problem in order to get an efficient solution*)

   The ministers of the cabinet were quite upset by the message from the Chief of Security stating that they would all have to change the four-digit room numbers on their offices.

   - It is a matter of security to change such things every now and then, to keep the enemy in the dark.
   - But look, I have chosen my number 1033 for good reasons. I am the Prime minister, you know!
   - I know, so therefore your new number 8179 is also a prime. You will just have to paste four new digits over the four old ones on your office door.
   - No, it's not that simple. Suppose that I change the first digit to an 8, then the number will read 8033 which is not a prime!
   - I see, being the prime minister you cannot stand having a non-prime number on your door even for a few seconds.
   - Correct! So I must invent a scheme for going from 1033 to 8179 by a path of prime numbers where only one digit is changed from one prime to the next prime.

   Now, the minister of finance, who had been eavesdropping, intervened.

   - No unnecessary expenditure, please! I happen to know that the price of a digit is one pound.
   - Hmm, in that case I need a computer program to minimize the cost. You don't know some very cheap software gurus, do you?
   - In fact, I do. You see, there is this programming contest going on...

   Help the prime minister to find the cheapest prime path between any two given four-digit primes! The first digit must be nonzero, of course. Here is a solution in the case above.
   1033
   1733
   3733

3739
3779
8779
8179

The cost of this solution is 6 pounds. Note that the digit 1 which got pasted over in step 2 can not be reused in the last step – a new 1 must be purchased.

Miscellaneous Problems

1. **Finding a sink vertex**
   Let $G = (V, E)$ be a directed graph on n vertices given in the form of an adjacency matrix M. A vertex $x \in V$ is said to be a sink vertex if for every other $y \in V \backslash \{x\}$, $(y, x) \in E$ and $(x, y) \notin E$. Design an $\mathcal{O}(n)$ time algorithm to report a sink in G, if exists.
   Hints: Note that you can afford to explore only $O(n)$ entries of M to solve the problem. How can exploring an entry of M be useful ? Can you determine in $O(n)$ time if a given vertex is a sink ?

2. **Coloring of a graph** A coloring of a graph $G = (V, E)$ is an assignment of colors to the vertices V such that for each edge $(u, v) \in E$, color assigned to u is different from the color assigned to v. A graph is said to be k-colorable if it can be colored using at most k colors. What is the smallest value k such that each graph on n vertices is k-colorable ? Design an efficient algorithm to determine if a given graph on n vertices is 2-colorable. What can be the maximum number edges in a graph on n vertices which is 2-colorable ?