

ESO207: Programming Assignment 2

Due on 13 Nov, 2015. To be submitted online.

Problem Given a graph G and a start-vertex s , perform BFS and compute the distance of each vertex from s .

Input:

- First input will be the number of vertices n . The vertex set will be assumed to be $\{1, 2, 3, \dots, n\}$.
- Second input will be the number of edges m .
- Suppose the edges are $\{n_1, n_2\}, \{n_3, n_4\}, \dots, \{n_{2m-1}, n_{2m}\}$, then the programme will expect $2m$ numbers, $n_1, n_2, n_3, \dots, n_{2m}$, as input.

Task 1 Construct the adjacency list (linked list) for each vertex and an array $g[]$ such that $g[1]$ points to the list of vertex 1, so on. Note that each list must be sorted in increasing order. This is required because we want to visit the neighbors of any vertex in increasing order in the DFS. No other order will be accepted.

Task 2 Implement BFS algorithm described in the class.

Task 3 Add one more task in your program. First time when the algorithm finds that it has reached a vertex already in "current" state then you do the following: suppose this happens when vertex u is being expanded and a neighbor v is extracted from the adjacency list of u which is in "current" state and it is different from $parent(u)$, then return two paths: first $u, parent(u), parent^2(u), \dots, s$ and next in a fresh line $v, parent(v), parent^2(v), \dots, s$. This is to be done only once. Clearly, if the input graph is a tree, then this task will not be executed.

Note: $\{uv\}$ is an edge in the graph which is not an edge of the BFS tree. This edge along with the two paths mentioned above form a cycle in the graphs.