

COSC2430: Programming and Data Structures

Homework 5: Graph Algorithm using Queue

1 Introduction

Reachability from a source vertex s is the problem to find the set of vertices S such that $v \in S$ if exists a path from s to v . You will create a C++ program to find reachable vertices using **Breath First Search (BFS) algorithm**.

2 Input and Output

$G = (V, E)$ is a directed graph, where n vertices and m edges. G can be represented as an adjacency matrix $E, n \times n$, where $n \leq 100$. Please see Figure 1 as an example. You will read a sparse matrix E from an input file; There will be ONE matrix entry per line in the input file and each line will have a triplet of integer numbers i, j, v where $1 \leq i, j \leq n$ indicate the entry and $v \neq 0$ indicates a directed edge pointing from vertex i to j . Given a source vertex $1 \leq s \leq n$, your program should display all reachable vertices (**excluding the source vertex s**) in **ascending order**.

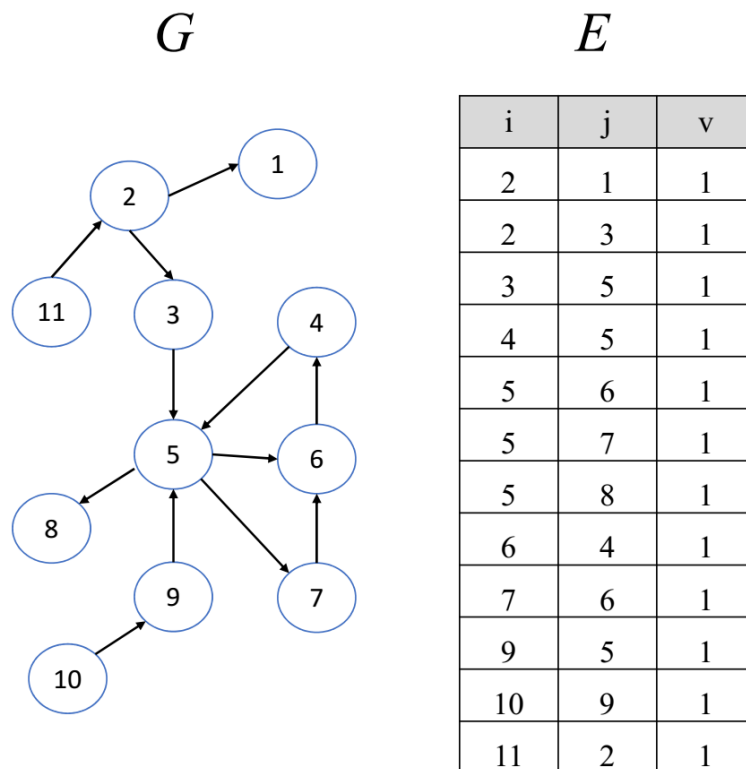


Figure 1: The adjacency matrix E (sparse representation) for a sample graph G

Input example for Figure 1 (the last line indicates the matrix dimension $n = 11$).

```
2 1 1
2 3 1
3 5 1
4 5 1
5 6 1
5 7 1
5 8 1
6 4 1
7 6 1
9 5 1
10 9 1
11 2 1
11 11 0
```

Output example (source=2)

```
1 3 4 5 6 7 8
```

3 Program and input specification

The main program should be called "reachability". Call syntax:

```
reachability "E=<input_file>;source=<num>".
```

4 Requirements

- `std::queue` or `std::deque` is not allowed to be used. You should implement your own queue data structure.
- Your program should get the result within 10 seconds.