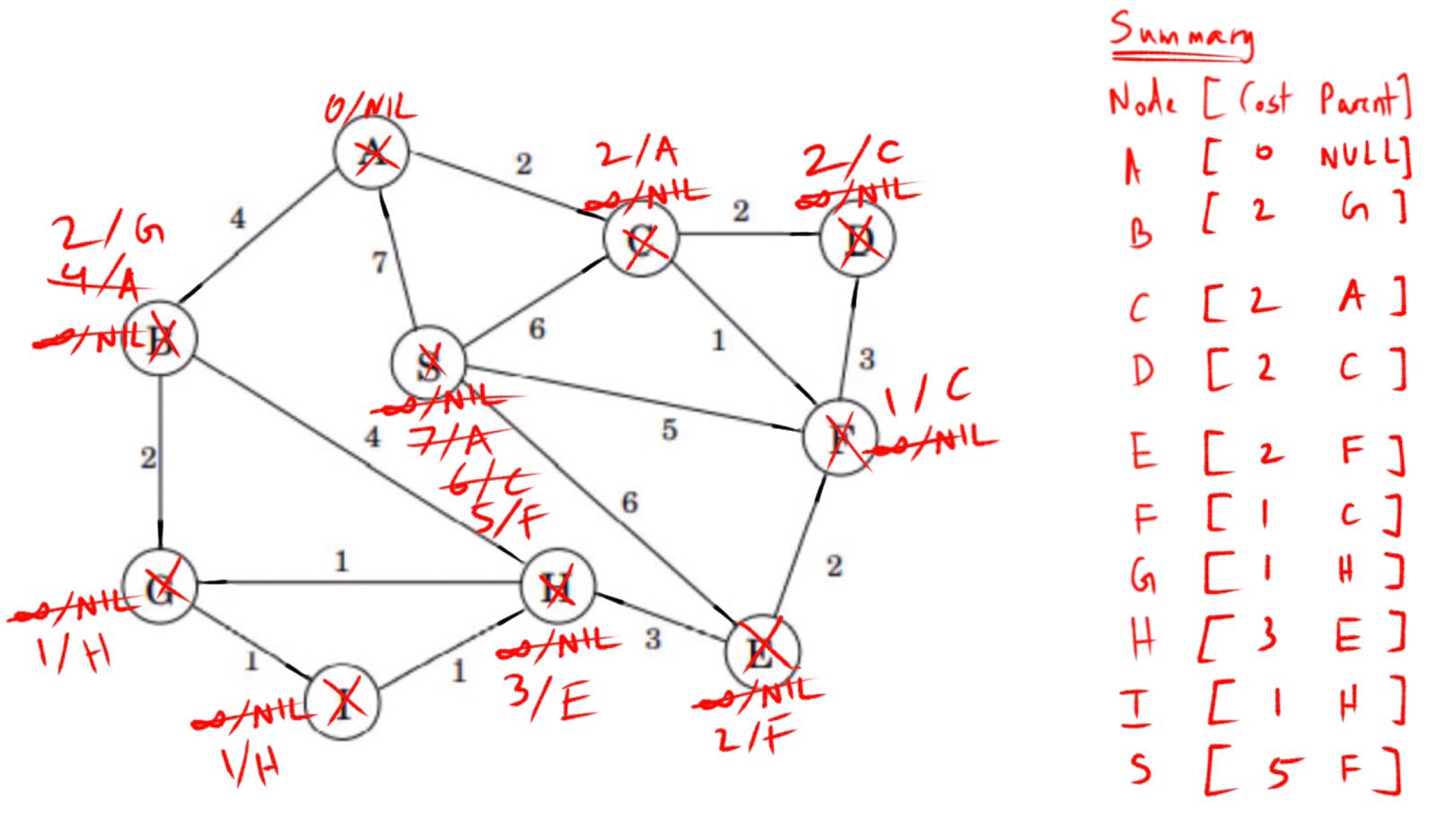
**Muhammad Anas Tanweer 22L-7790 BCS-4C**

**Design & Analysis of Algorithms Assignment 4**

**Problem 1 - Dry Runs**

**Question 1a,** code also available in **Project86.cpp**



Minimum spanning tree cost = Σ(cost) = 0 + 3(1) + 4(2) + 3 + 5 = **19**

**Question 1a END**

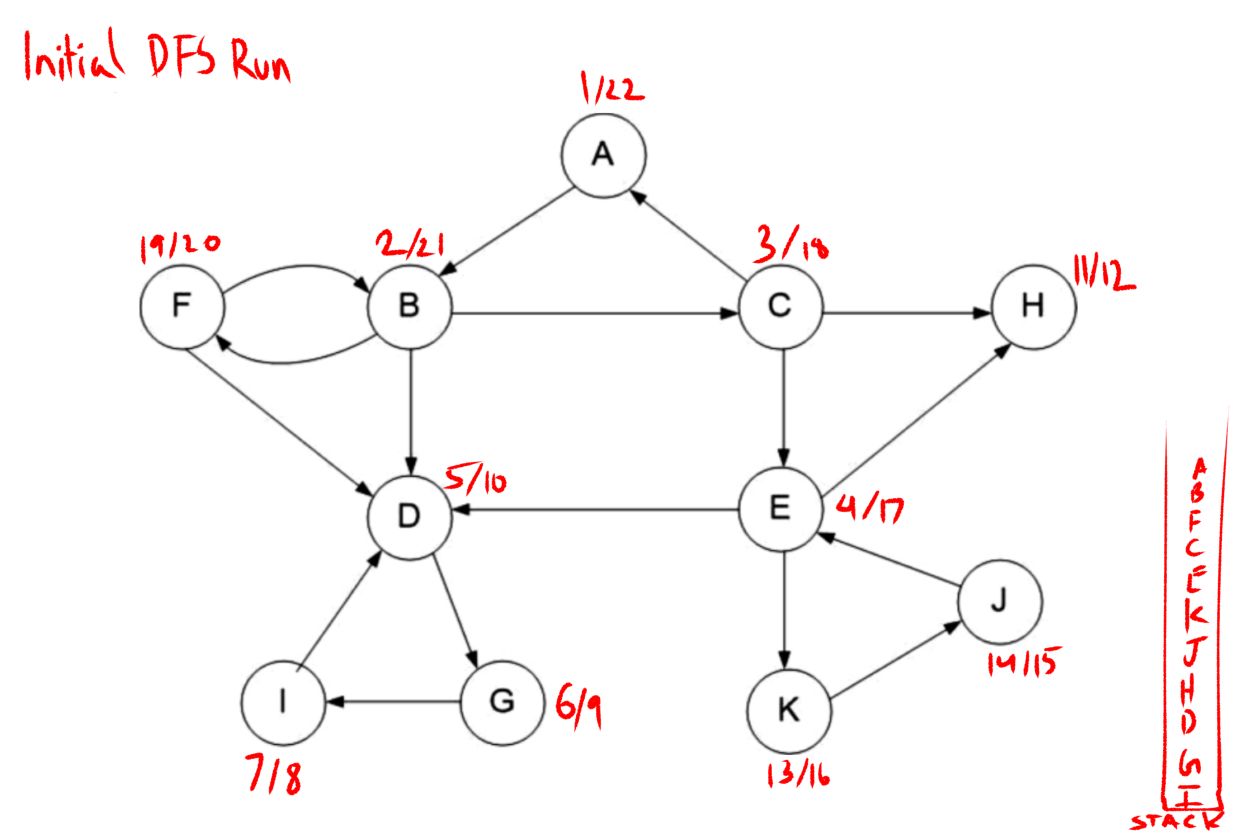
**Question 1b,** A diagram of a diagram with red text

Description automatically generated with medium confidence

Minimum spanning tree cost = Σ(weight) = 3(1) + 4(2) + 3 + 5 = **19**

**Question 1b END**

**Question 2,** code also available in **Project80.cpp**



A diagram of a network

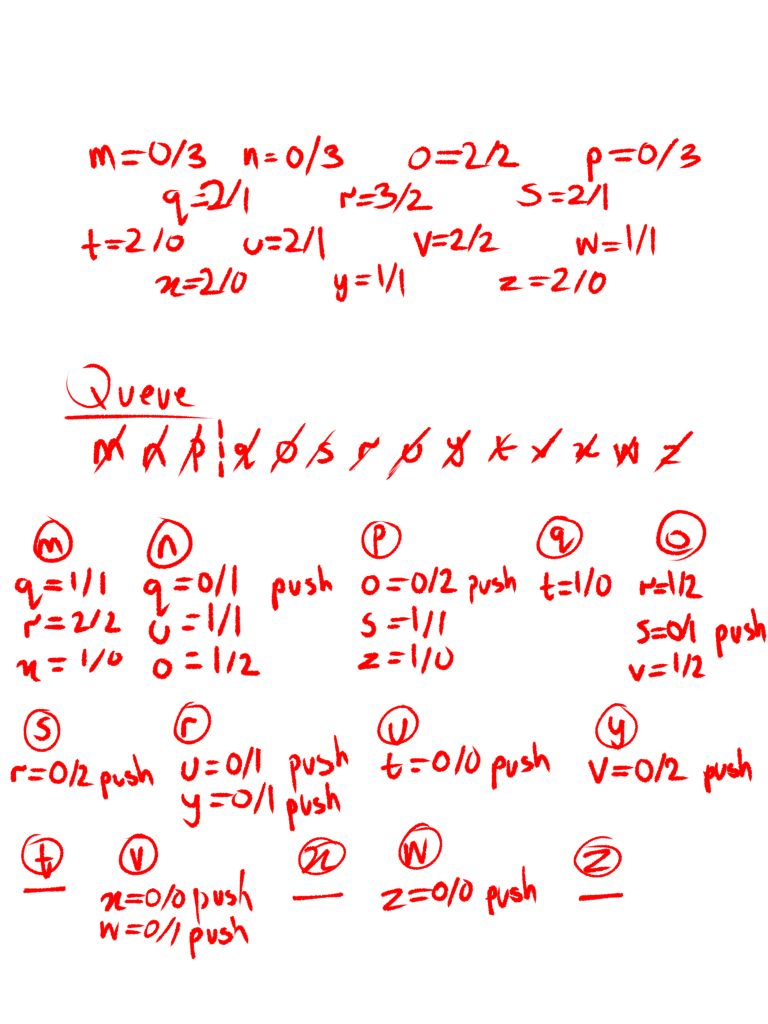
Description automatically generated

A diagram of a diagram

Description automatically generated

**Question 2 END**

**Question 1,**

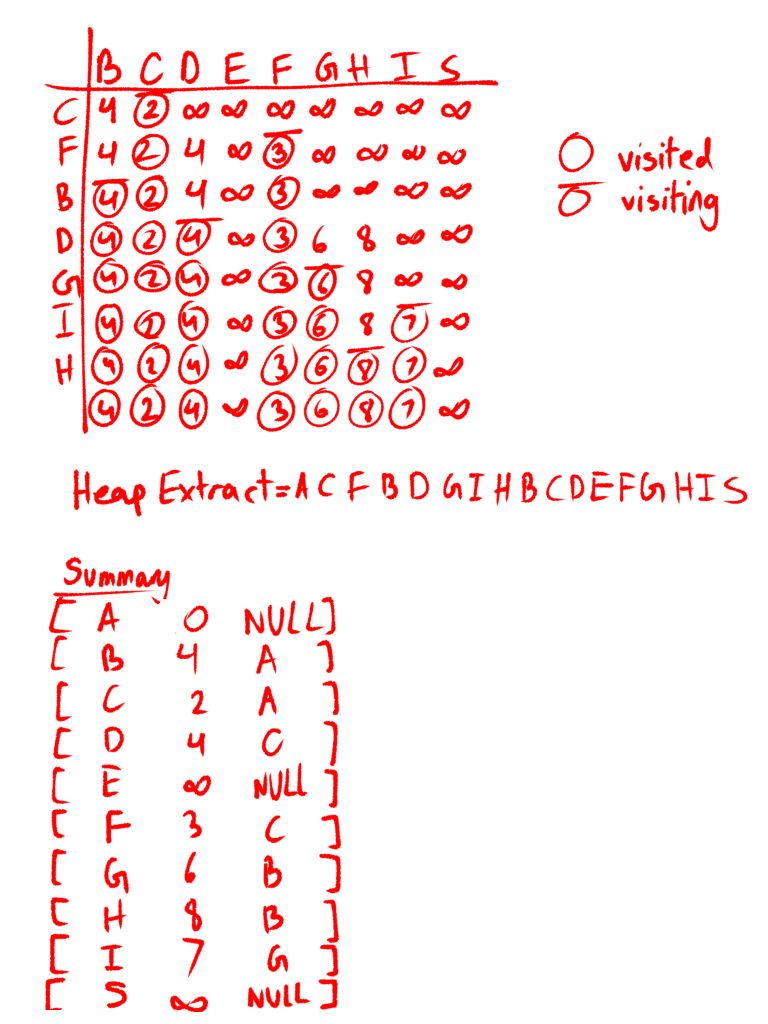


A diagram of a diagram of a diagram

Description automatically generated with medium confidence

**Question 1 END**

**Question 3,** code also available in **Project87.cpp**



**Question 3 END**

**Question 4,** code also available in **Project87.cpp**A white board with red writing

Description automatically generated

A red marker on a white sheet

Description automatically generated with medium confidence

**Question 4 END**

**Problem 2 – Design**

**Question 2,** code in **Project85.cpp**

**Question 3,** code in **Project90.cpp**

**Question 4,** code in **Project80.cpp** & **Project84.cpp**

**Project80.cpp** is the implementation of Kosaraju’s Algo

**Project84.cpp** the implementation of Professor Bacon’s Hypothesis

Both give the same result

**Question 5,** code in **Project88.cpp**

The only addition made to Dijkstra Algo was edgeCost. Now the “if” statement also compares and relaxes on the basis of edgeCost (similar logic to minimizing weight cost)

**Question 6,** code in **Project89.cpp**