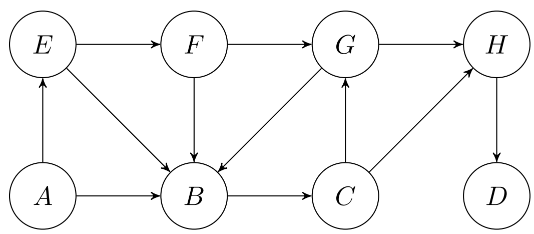
Weekly Assignment 2:
Breadth-First Search

September 2023

1. Run the BFS algorithm on the directed graph below, using vertex A as the source. Show all distances and the BFS tree.



2. Show how to implement a queue using two stacks. Analyze the running time of the queue operations.
3. Let us consider the following (incomplete) program:

Algorithm 2 Incomplete program, stack as array of strings.

```
stack: array of strings
no_elements: integer

//creates empty stack
procedure CREATESTACK()
    no_elements ← 0
    SetLength(stack, no_elements)        ▷ sets size of the array of strings "stack" to "no_elements"
end procedure

//tests if the stack is empty (may return true of false)
function ISEMPTY()
    xxx TODO xxx
end function

//pushes "element" to the stack
procedure PUSH(element)
    xxx TODO xxx
end procedure

//removes and returns last element on stack
function POP()
    xxx TODO xxx
end function

//returns top element
function TOP()
    xxx TODO xxx
end function

//displays elements in the stack
procedure DISPLAY()
    xxx TODO xxx
end procedure

createStack()
push("AAA")
display()                                ▷ 1
push("BBB")
push("CCC")
display()                                ▷ 2
name ← pop
WriteLn(name)
display()                                ▷ 3
```

(a) Complete the functions and procedures above.

(b) Give the complexity of each function.

(c) What is displayed on the screen?[1]

(d) Write a function *empty*(), emptying the stack.

4. Given a graph $G = (V, E)$, a 2-coloring is a function $c\colon V \to \{blue, red\}$ assigning colors to vertices such that $(u, v) \in E$ implies $c(u) \neq c(v)$, i.e., adjacent vertices have different colors. Let G be an undirected

[1] recall that `WriteLn(variable)` writes the content of `variable` in the terminal.

graph. An odd-length cycle is a sequence $v_1, v_2, \ldots, v_k$ of vertices of G such that $(v_i, v_{i+1}) \in E$, for $i = 1, \ldots, k-1$, $(v_k, v_1) \in E$ and k is odd.

1. Show that, if G has an odd-length cycle, then there is no 2-coloring.
2. Assume G has no odd-length cycles. Use BFS algorithm to find an appropriate 2-coloring for G and argue that it is correct. Hints:
   (a) assume that your algorithm does not produce a valid coloring, i.e., there is a pair of adjacent vertices with the same color, and prove the claim by contradiction;
   (b) prove and use the following fact: two vertices with the same color must both have even or odd shortest distance from the BFS source.

5. Together with your friend Eager Dukstra, you are developing an app that new students may use to compute the shortest walking route between any pair of locations on the campus of Radboud University. You already have constructed an undirected graph $G = (V, E)$, where V is large collection of points of interest on the campus, and a set of edges E connecting these points. You also have a weight function $w : E \to \mathbb{N}^{>0}$ that specifies for any edge $(u, v) \in E$ the distance (in meters) between the corresponding points of interest. For each path $p = (v_1 v_2 \cdots v_k)$ of G, the weight of p is simply the sum of the weights of the edges contained in it, that is, the distance from v_1 to v_k when you follow path p:

$$w(p) \quad = \quad \sum_{i=1}^{k-1} w(v_i, v_{i+1}).$$

Eager claims that, once the starting location s is known, we may use a simple adaptation of the Breadth-First Algorithm to compute the path with mininal weight (minimal length in meters) from s to any other point of interest on the campus.

Compared to the implementation given in the slides (slide 37, "lecture2.pdf"), one has to keep track of an array d where $d[w]$ represents the distance from s to the vertex w. This array is initialized to be 0 for every vertex, and in after line 18, one adds

$$d[v] \leftarrow d[u] + w(v, u).$$

Note: there are two versions of the slides on Brightspace, containing a slightly different version of breadth-first search. The previous description was based on the slides with the name "lecture2.pdf". If one uses the slides with the name "BFS.pdf", then one only has to change line 15 on slide 26 with the BFS pseudocode from

$$d[v] \leftarrow d[u] + 1$$

to

$$d[v] \leftarrow d[u] + w(v, u).$$

What do you think of Eager's proposal? Argue why it is correct or provide a counterexample.