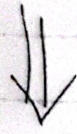
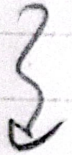


Focus is 2 Things

□ How BFS Should Look Like



Info Displayed



- 1) Queue
- 2) Vis Nodes

a. User Chooses Start Vertex

b. BFS Visualizer Starts running

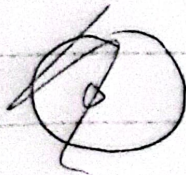
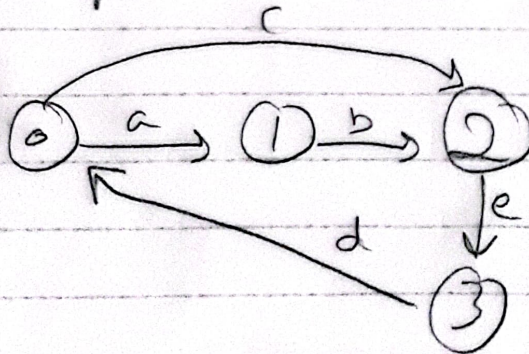
- The Visited Nodes are Colored "Green"

- The edge used is Colored

- The edges we traverse are Made Bold

- The Current Node are Colored "Blue"

- The unvisited Nodes Don't change



BFS(0)

~~0~~, colored blue

c, colored red

2, colored green

c, made bold

0, colored red

1, colored green

a, made bold

2, colored green

1, colored blue

When it comes
the case

1 → 2

b will be

Colored red

but never Made

bold

2] Hard Design & Implementation Tasks:

1. How will we log Data for all Visualizers

2. How will you Stop Drawing

- any other process can stop Drawing
- What if during visualization User enters new data

⇒ 1 idea Have 2 Btns representing different Modes

Drawing	Visualizing
---------	-------------

→ When user enters data he is in drawing

→ When user chooses one of the Visualizers the ~~Page~~

- Drawing is Freezed
- The Visualizing Mode Btn is highlighted
- The Text Area is greyed out

→ When user clicks the Drawing Btn the

- the text area is not greyed out

- Visualization stops if one was in progress
- All graph colors are reset
- The dropdown Visualizer is set to select

3. How Do we change the Node and Edge Colors

- Challenge 1 how to change Colors of Nodes and Edges that are stored inside the Graph Obj

+ How to make this process as Abstract as possible for any Visualizer

- How can other Visualizers rest to default Color Settings

3) Some Ideas:

- Split the Graph Reading and Data Structure into Separate Class from the Graph Drawing

Graph | GraphDrawer

- Separate the Classes based on if it is a class related to UI or Non-UI

- Add 2 Separate Modules

- User Input Module
- Visualization Module

+ contains a class for each Graph Visualizer

- Can Creating Multiple Instances of GraphUI be useful

- Is Separating the Visualization Algorithm and Visualization UI useful, using for example something like the Observer Pattern

+ Probably a Slow IDEA