FIFO Queue (I): Introduction

CSCD 300 – Data Structures

Eastern Washington University

EASTERN WASHINGTON UNIVERSITY
start something big

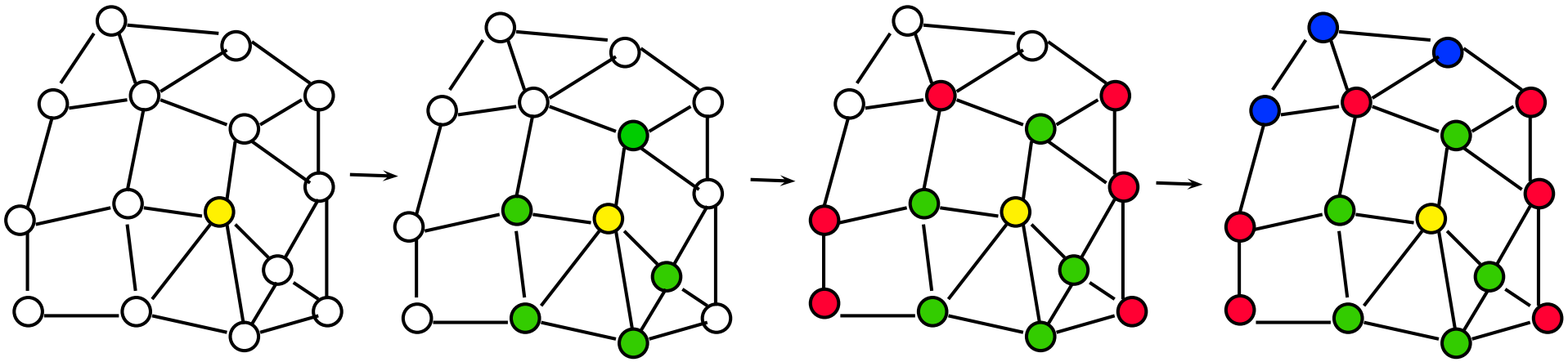
Goal

We will learn the conceptual view of a new data structure: the First-In-First-Out Queue.

Outline

An example application: measuring the "social distance"

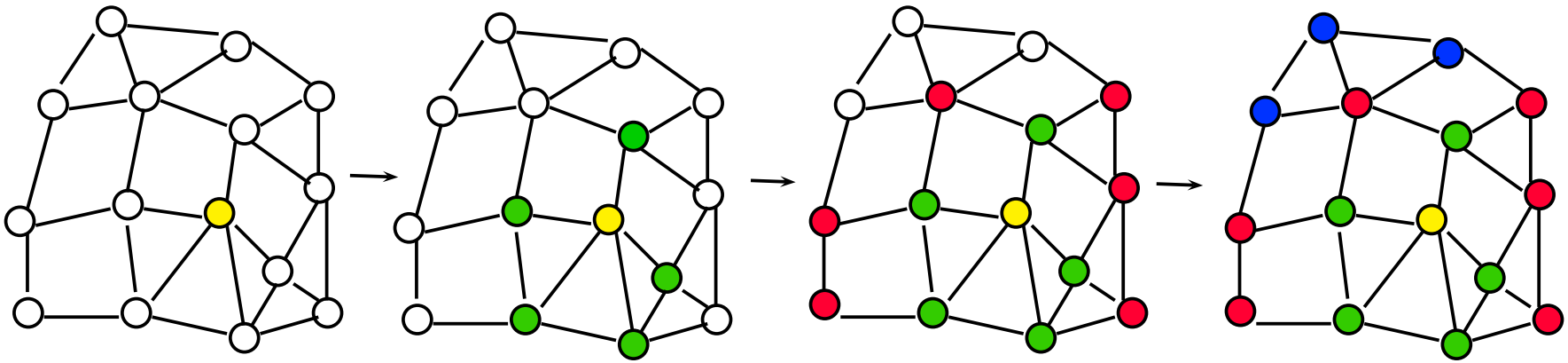
Task: You (the yellow node) are in a social network as shown below. You want to print all the people, with whom you have either direct or indirect connections, in the non-increasing order on how "close" they are.



How to implement the above printing ?

Use a first-in-first-out queue, similar to those at Walmart [1]

```
Initialize an empty queue.
Insert the yellow node into the queue.
While(the queue is not empty){
   1) Print and discard the node at the head of queue.
   2) Insert all those nodes, which are directly connected to the
      previously discarded head node and were not inserted into
      the queue before, into the tail of the queue.
}
```



Note: trace the queue by yourself by following the pseudocode above.

[1] The algorithmic idea is called graph's breadth first search, which will be discussed in more details in CSCD320.

The First-In-First-Out Queue (FIFO Queue)

The FIFO queue is a conceptual computer science data structure. It can be viewed as a container for storing data elements and follows the first-in-first-out order:

- When a new element e arrives and wants to join the queue, it is always inserted into the tail side of the queue. This action is often called: `enqueue(e)`.
- When we need to delete an element from the queue, we always delete the one from the head side of the queue. This action is often called: `dequeue()`.

The FIFO queue data structure can be implemented using the array or linked list data structure. We will discuss both implementations in the following lectures.

(See the attached Java source code for the interface of the FIFO queue data structure.)