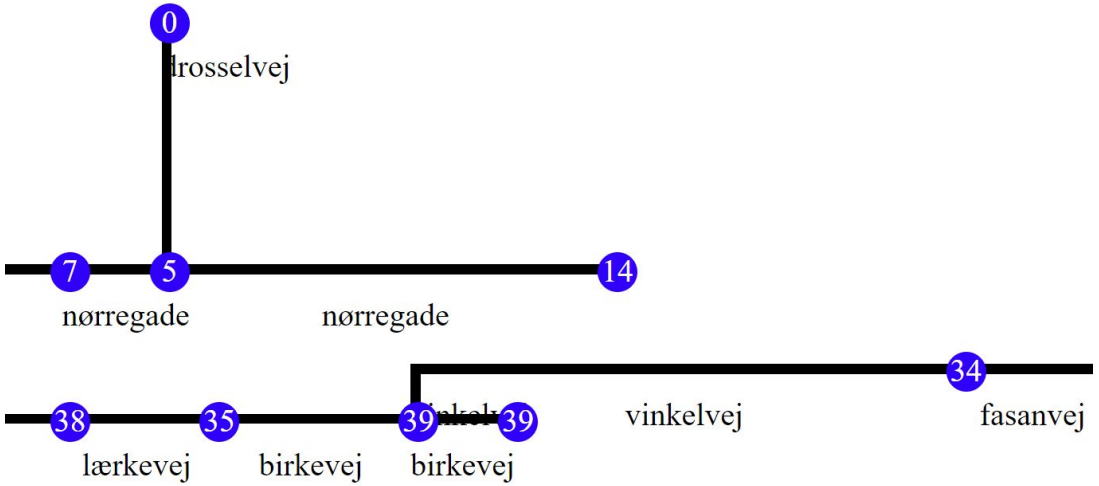
Navigation Application Group Project

Algorithms and Discrete Maths Spring 2023

Mads, Malthe, Mathias & Tobias

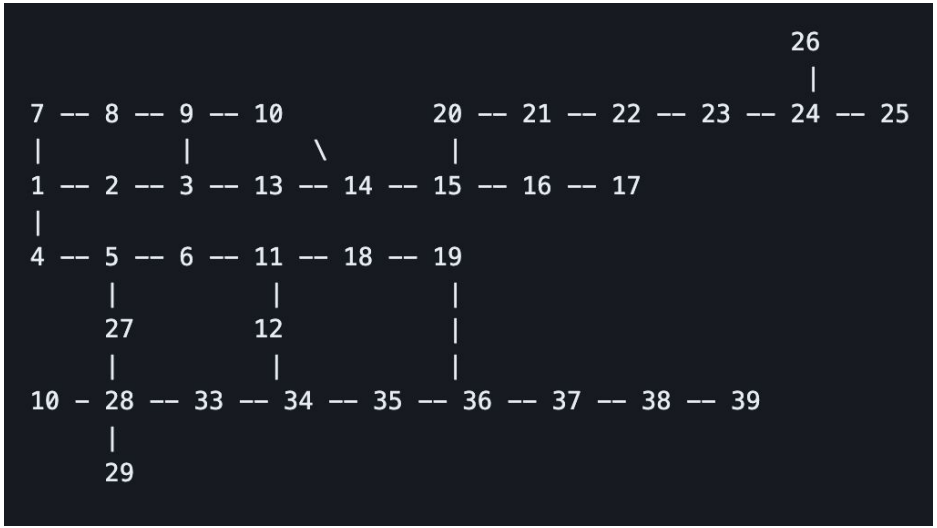
Interface

- Introduction
- Demo
- Docker + NGINX
- Dijkstra's shortest path



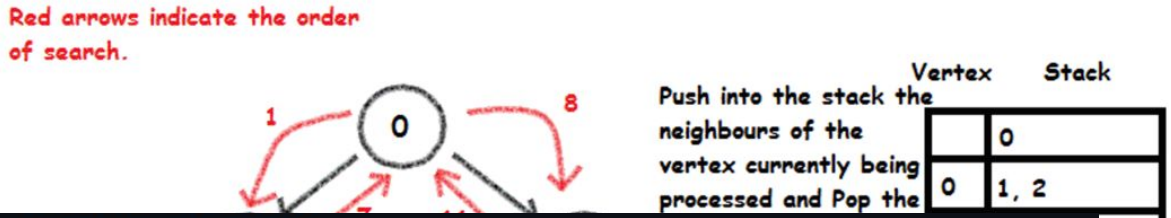
Data

- Not necessarily created for graphs
- Data is put into postgres
- 3 tables:
- road, road_part, road_part_relation
- Made into a graph after



Depth-First Search

- Recursive algorithm
- From root to edge
- Stack data structure

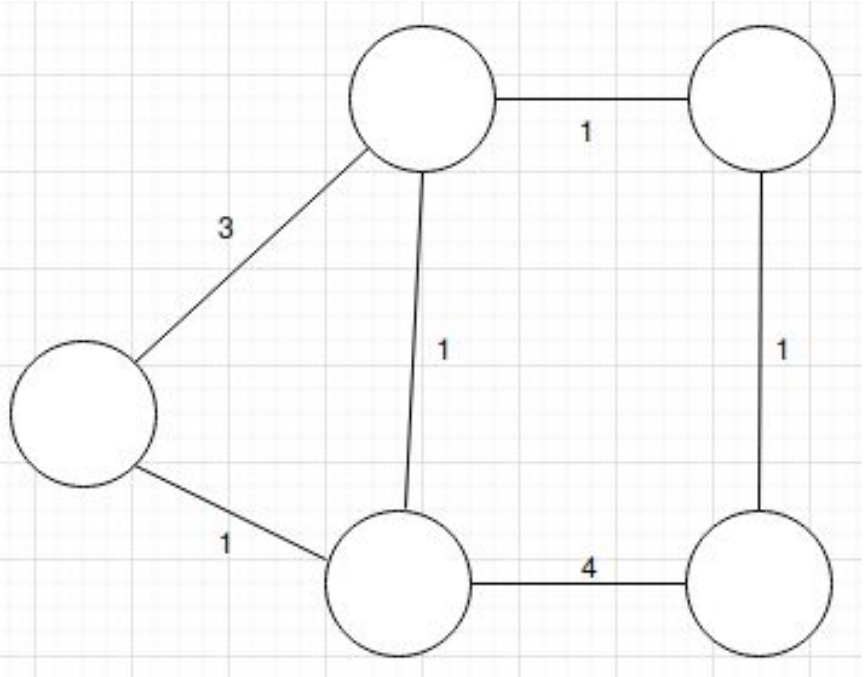



```rust
#[actix_rt::test]
async fn can_find_node_with_dfs() {
    with_correct_env(async {
        let map = Map::new().await.expect("Map was not found");
        let val = map.find_roadpart_id("skovvej");
        assert!(val.is_some())
    })
    .await;
}
```

- Advantages
- Disadvantages
- Complexity

Dijkstra's Shortest Path

- Path finding algorithm
- Shortest path to anywhere



Dijkstra's Shortest Path

- Example usage
- Previous nodes

```
async fn weights_should_be_added_together() { ▶ Run Test | Debug
    with_correct_env(async {
        let map = Map::new().await.expect("Map was not found");
        let nodes = map.shortest_path(&1).unwrap();
        assert_eq!(nodes.get(&3).unwrap().borrow().node.id, 3);
        // // 0 + 1 + 2
        assert_eq!(nodes.get(&3).unwrap().borrow().weight, 3);
    })
    .await;
}
```