

# Generating brain-like networks with distance-dependent connection probability

Patrik Broček

Rastislav Nowak

Martin Oravec

## Task 1

### Task 1.1

TODO pokec

```
SEED = 42
generator = np.random.default_rng(SEED)
rng = lambda: generator.random()
dist = lambda n1, n2: np.sqrt((n1[0] - n2[0])**2 + (n1[1] - n2[1])**2)
p = lambda d, alpha, beta : beta * np.exp(-alpha * d)

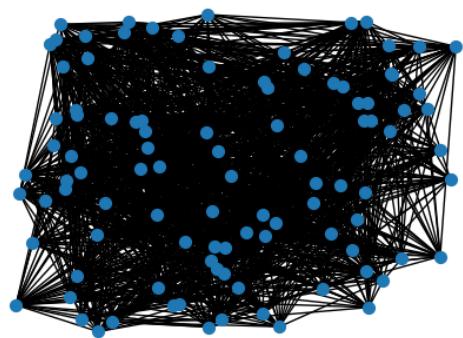
def KaiserHilgetag(alpha = 1, beta = 1, N = 100):
    adjacency = np.zeros((N, N))
    neurons = [(rng(), rng())]
    while(len(neurons) < N):
        new_neuron = (rng(), rng())
        dists = [dist(new_neuron, n) for n in neurons]
        neighbours = [i for i, d in enumerate(dists) if random.random() < p(d, alpha, beta)]
        if len(neighbours) > 0:
            neurons.append(new_neuron)
            for i in neighbours:
                adjacency[i, len(neurons)-1] = 1
                adjacency[len(neurons)-1, i] = 1

    return adjacency, neurons
```

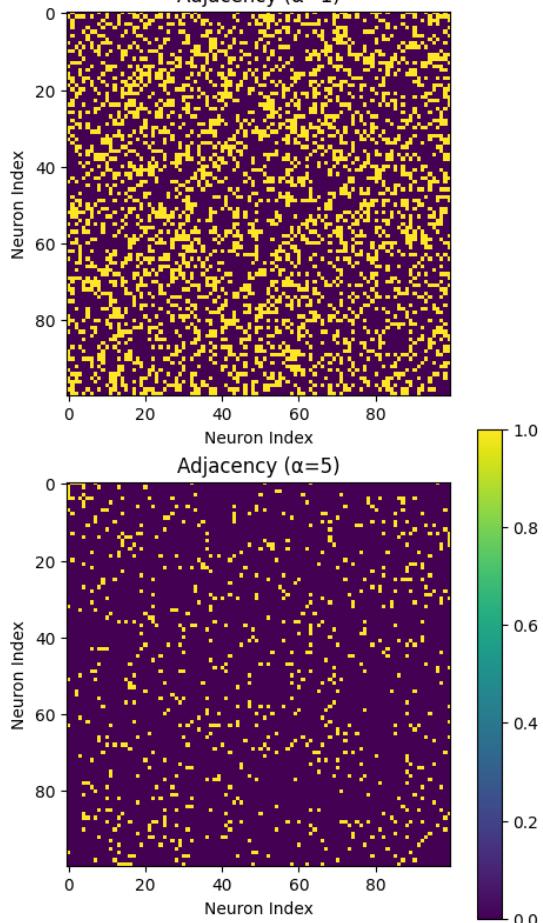
## **Task 1.2**

TODO pokec

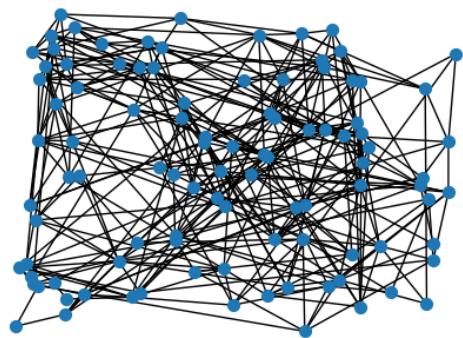
Graph ( $\alpha=1, \beta=0.5$ )



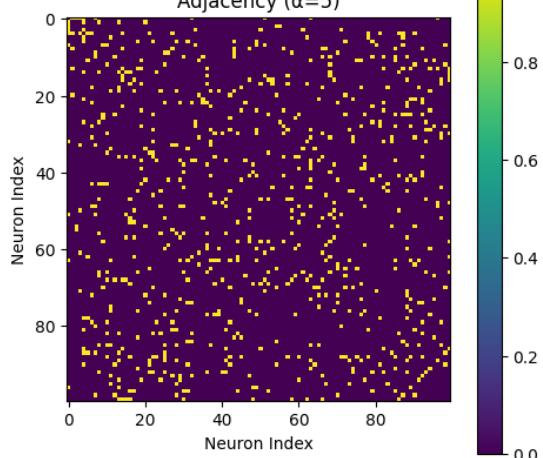
Adjacency ( $\alpha=1$ )



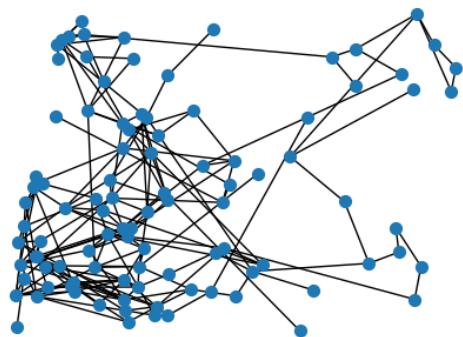
Graph ( $\alpha=5, \beta=0.5$ )



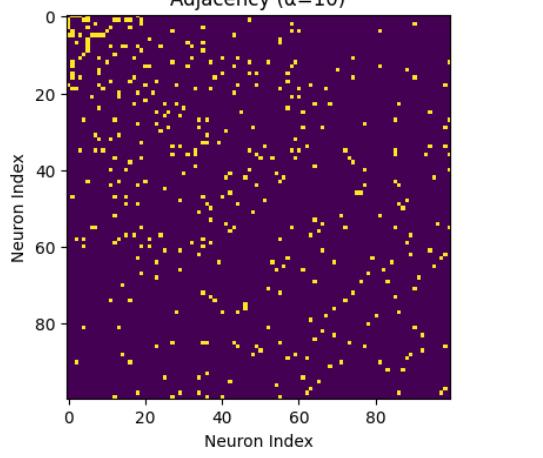
Adjacency ( $\alpha=5$ )

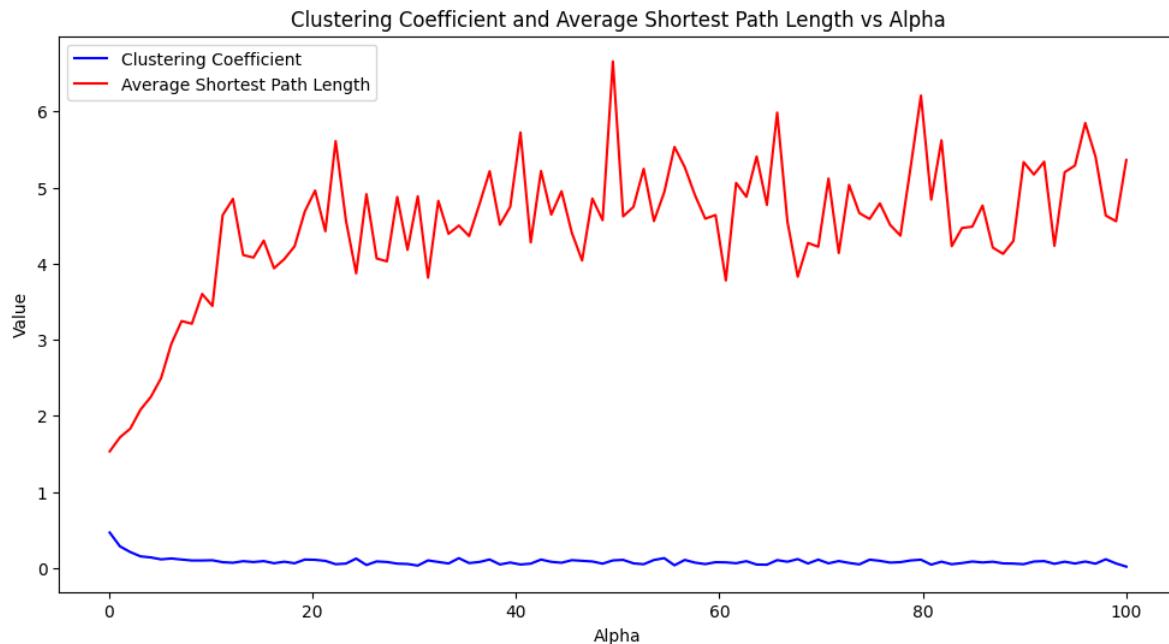


Graph ( $\alpha=10, \beta=0.5$ )



Adjacency ( $\alpha=10$ )





Alpha values yielding desired properties: []

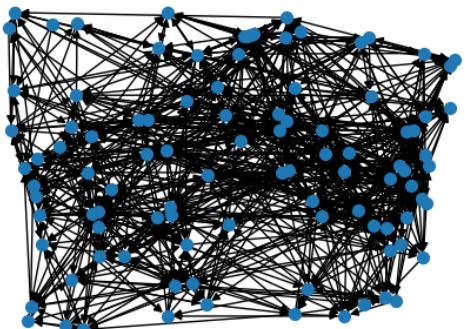
## Task 2

### Task 2.1

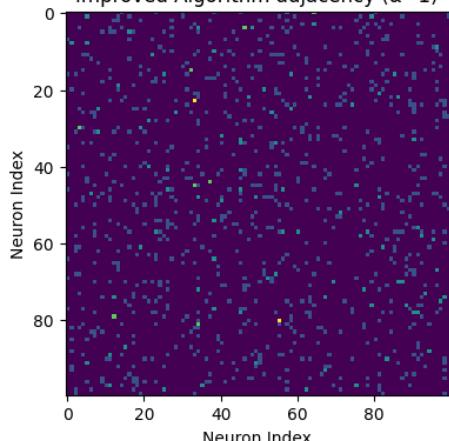
### Task 2.2

TODO pokec alebo neviem

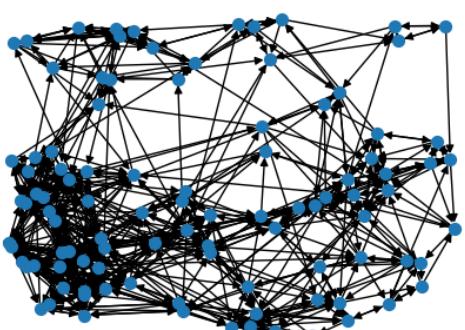
Improved algorithm graph ( $\alpha=1$ )



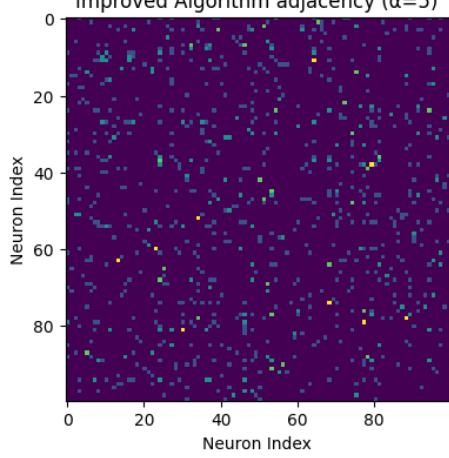
Improved Algorithm adjacency ( $\alpha=1$ )



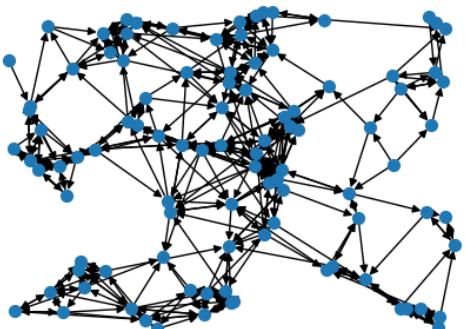
Improved algorithm graph ( $\alpha=5$ )



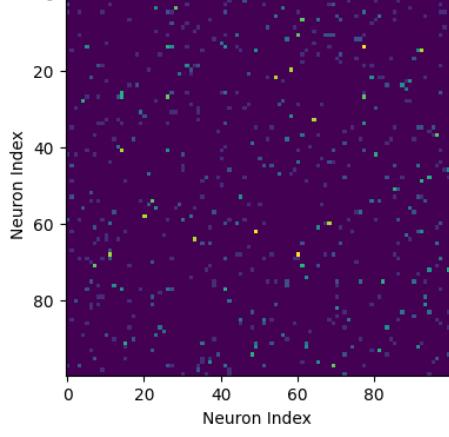
Improved Algorithm adjacency ( $\alpha=5$ )



Improved algorithm graph ( $\alpha=10$ )



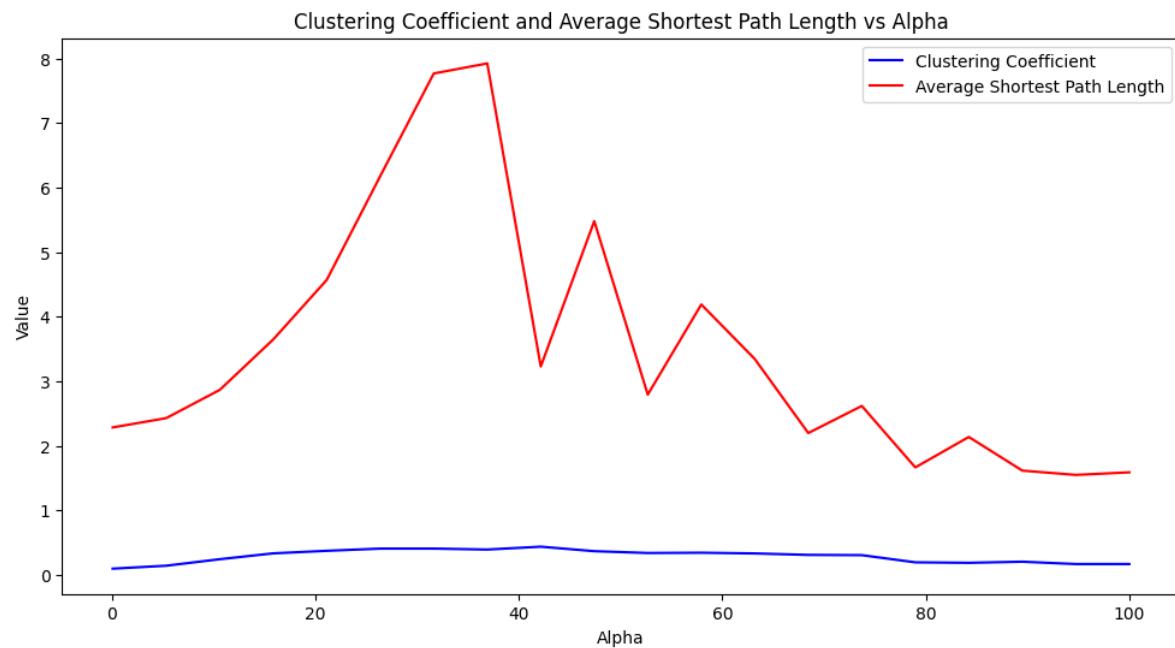
Improved Algorithm adjacency ( $\alpha=10$ )



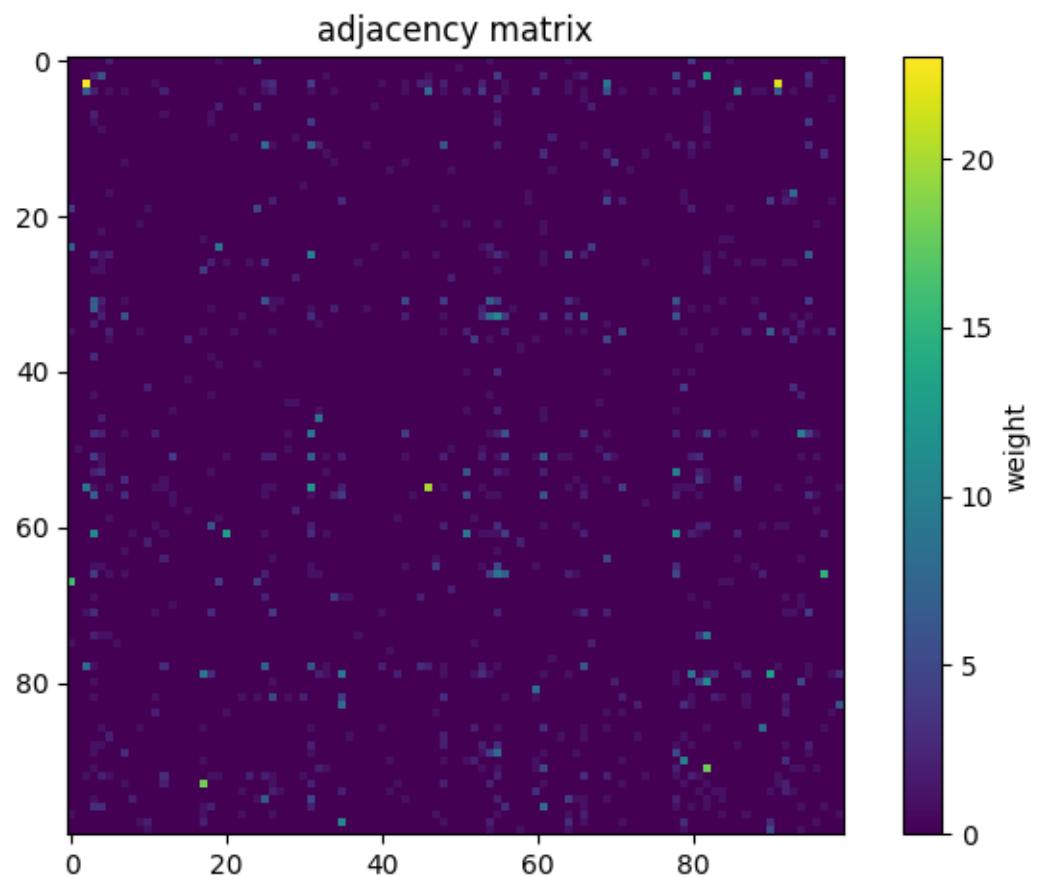
### Task 2.3

TODO pokec pokec pokec

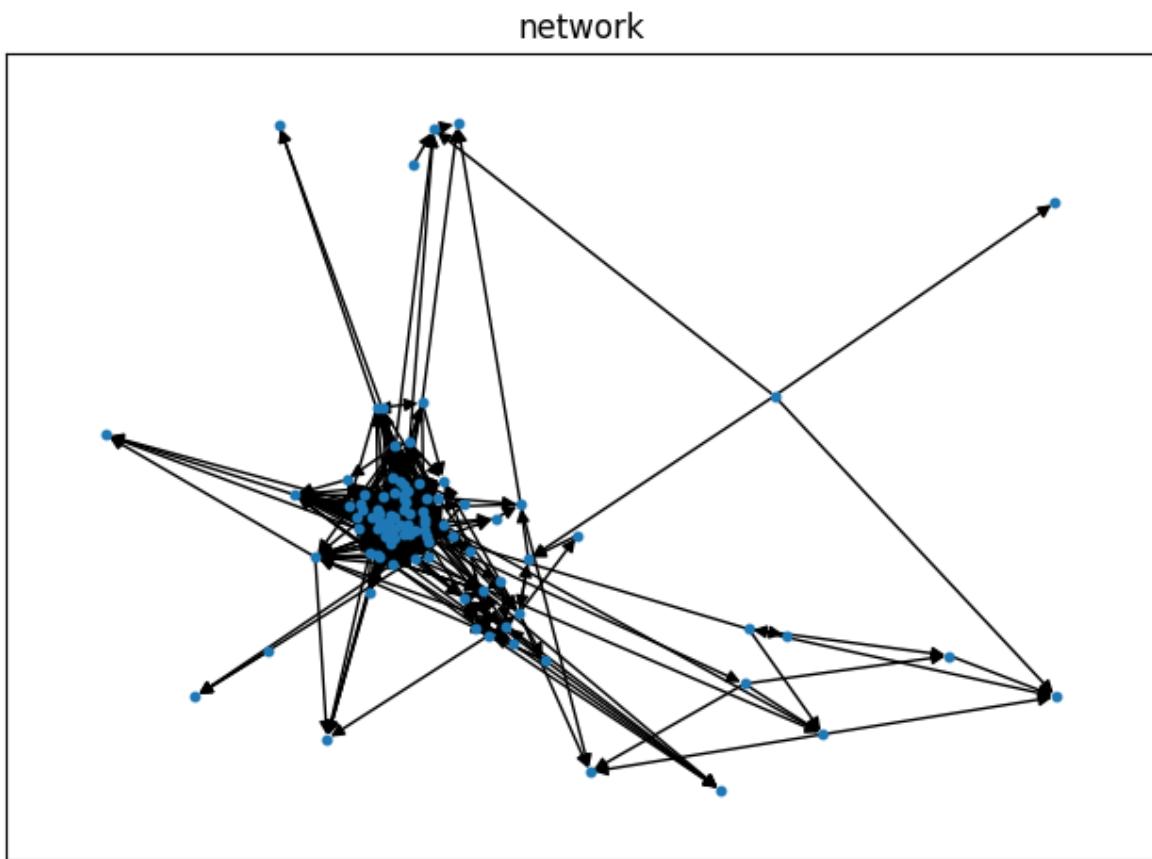
Calculating for alpha 20/20



### Task 3



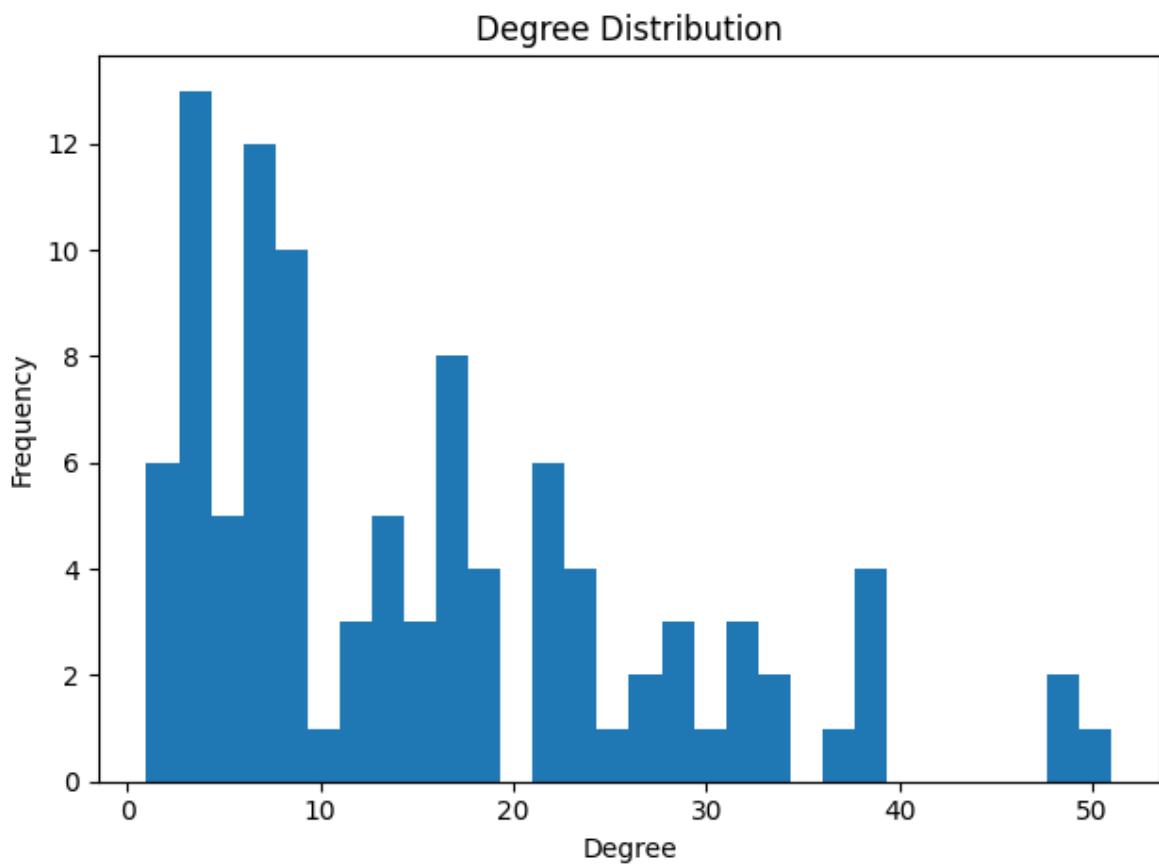
### Task 3.1



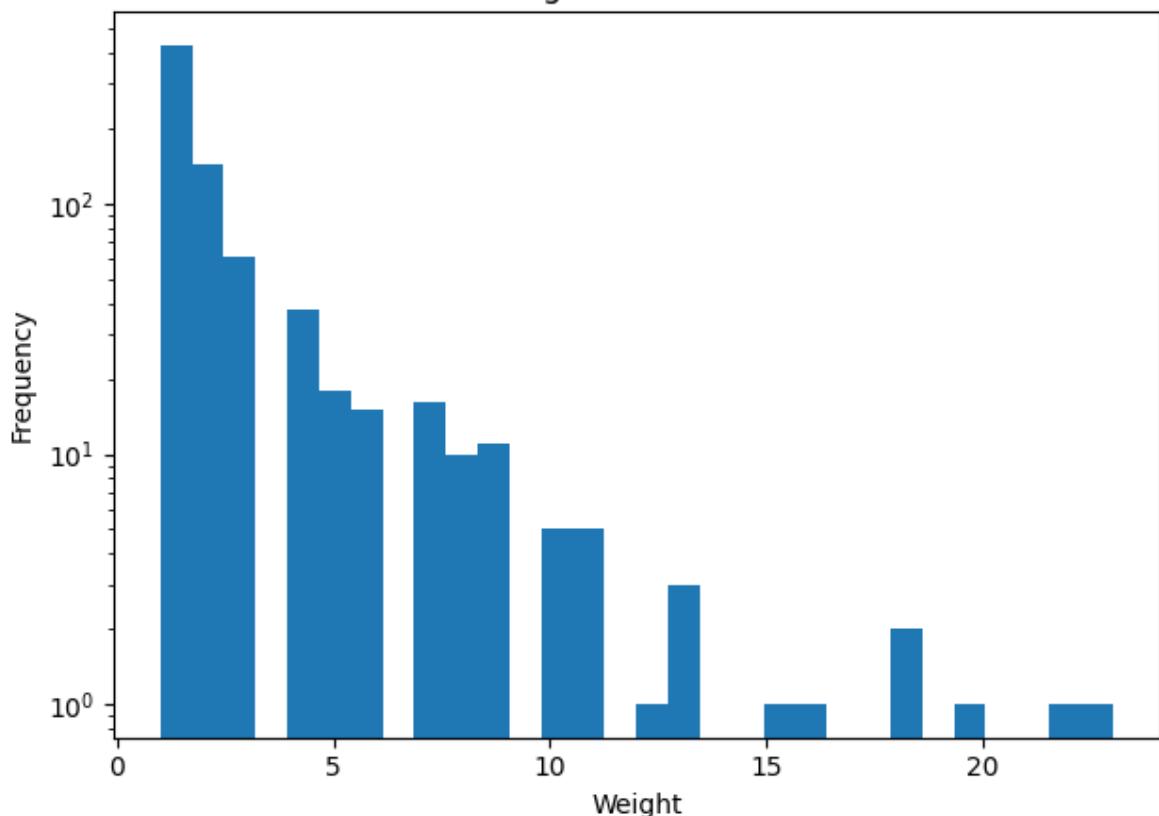
### Task 3.2

TODO ori dik

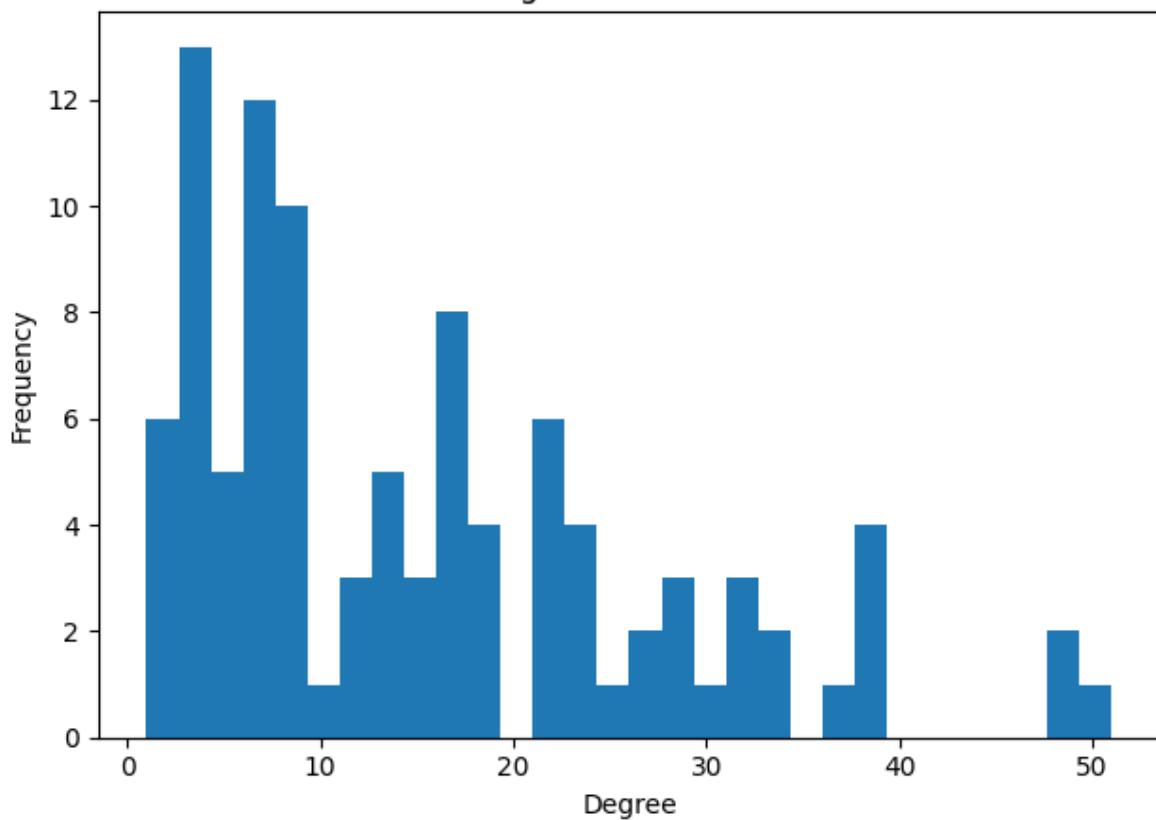
Avergae clustering is 0.2278 and average shortest path length is 2.6365

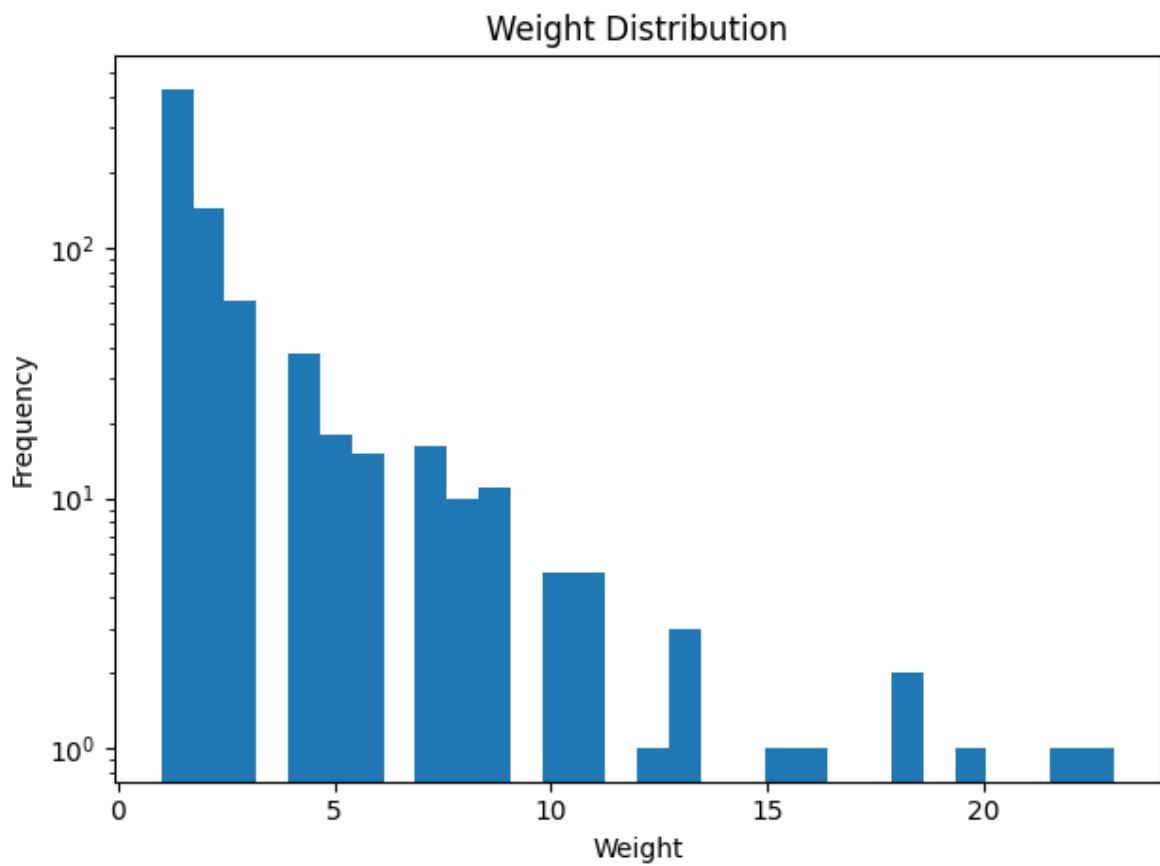


Weight Distribution



Degree Distribution





### 3.3

no budť tu len bude bunka, ktorú už som napísal alebo oraho prehrotene bin polenie