

# Programming Assignment

due April 27, 2021

1. Implement topological sort with DFS using adjacency lists.
2. Implement topological sort with priority heaps and in-degrees using adjacency lists.

The following code implements a random graph on  $n$  nodes and returns the result as an adjacency list.

```
import random

def create_random_graph(n=10, edge_prob = 0.1):
    result = [ [] for i in range(n) ]
    for i in range(n):
        for j in range(n):
            if not i == j and random.random() < edge_prob:
                result[i].append(j)
    return result

def pretty_print(adjacency_list):
    for i in range(len(adjacency_list)):
        print(i, adjacency_list[i])
```

Use 100 random graphs with 50 nodes and an edge-probability of 0.2 to measure the speed of both of your implementations and compare them. If you feel up to it, you can use statistics to decide whether your numbers are significant enough to justify saying that one implementation is better than another one.

Hand-ins:  
your code  
pdf (not docx, not excel) of your results.