Prof. Dr. Thomas Schultz
Olivier Morelle (morelle@uni-bonn.de)

B-IT
Bonn-Aachen International
Center for Information Technology
December 15, 2024

Winter term 2024/25

Computer Science for Life Scientists

Assignment Sheet 11

Solution has to be uploaded by January 8, 2025, 10:00 a.m., via eCampus

- This exercise can be submitted in **small groups** of 2-3 students. Submit each solution only once, but clearly indicate who contributed to it by forming a team in eCampus. Remember that all team members have to be able to explain all answers.
- Remember to include proper **documentation** in all your code, in particular, docstrings for functions.
- Please submit your answers as a single file in .ipynb or .pdf format.

If you have questions concerning the exercises, please use the forum on eCampus.

Exercise 1 (Disjoint Set Forests, *8 Points*)

In the lecture, we learned about disjoint set forests, a data structure that implements the disjoint set data type. We also learned about two heuristics that greatly improve their asymptotic running time: Union-by-rank and path compression. In this exercise, you will implement this data structure and experimentally observe the benefit from the heuristics.

a) Convert the pseudocode implementation of disjoint set forests that is given in the lecture slides into valid Python code. Use it to make 1.000 sets, and perform a random mix of 500.000 union and find operations on them, with random parameters. Report the resulting running time. (4P)

b) Remove the path compression. Repeat the experiment and report the resulting running time. (2P)

c) Now, additionally remove the union-by-rank heuristic. Instead, Link should always make the representative of the second set a child of the first set's representative. Repeat the experiment and report the resulting running time. (2P)

Exercise 2 (Minimum Spanning Trees, *8 Points*)

In the lecture, we introduced Kruskal's algorithm for finding Minimum Spanning Trees (MSTs). Consider the following alternative greedy algorithm that grows an MST of an undirected connected graph `G` with edge weights `w` starting at a given node `r`. This algorithm keeps all nodes `v` that are not yet part of the MST in a minimum priority queue with key `v.key`. When it terminates, an MST of `G` is given by the set of edges $\{(v.\pi, v) : v \in G.V - \{r\}\}$:

```
MST(G,w,r):
  for v in G.V:
    v.key = ∞
    v.π = None
```

```
5   r.key = 0
6   PriorityQueue Q = G.V
7   while Q!=∅:
8     u = Q.Extract-Min()
9     for v in G.Adj(u):
10      if v∈Q and w(u,v)<v.key:
11        v.π = u
12        Q.Decrease-Key(v, w(u,v))
```

a) What does it tell us about a node `v` $\in$ `Q` if, at the beginning of an iteration in line 7, `v.π!=None`? (1P)

b) Specify an invariant for the values of `v.key` for vertices `v` $\in$ `Q` with `v.π!=None` that should hold at the beginning of an iteration in line 7. Note that, in the pseudocode above, `Q.Decrease-Key(v,w)` sets `v.key=w` and updates the position of `v` in `Q` accordingly. (1P)

c) Use the invariant above to argue why it is safe to add the node `u` that we extract in line 8 to the MST via the edge $(u.\pi, u)$. (1P)

d) Does correctness of the result depend on the choice of starting node `r`? Could different starting nodes lead to different MSTs? Briefly justify your answers. (2P)

e) State the asymptotic running time of the given algorithm, and explain how you arrived at your answer. (2P) How does your result compare to the asymptotic running time we derived for Kruskal's algorithm? (1P)

Exercise 3 (Flow Networks, *9 Points*)

Do the following task manually and submit your sketches. You can make the figures on your computer or on paper. Make sure to include all figures in your final notebook.

- The following figure describes a flow network; all edges are labeled with their respective capacities. Apply the Ford-Fulkerson method given the augmenting paths in Table 1. For steps 1 to 4, write down the residual capacity and compute the current value of the flow. (4P)
- Draw the residual network at step 3 (i.e., after augmenting the path in step 3). (3P)
- Write down another augmenting path at step 5 and compute the corresponding residual capacity and the value of the flow. What is the value of the maximal flow in this network? (2P)

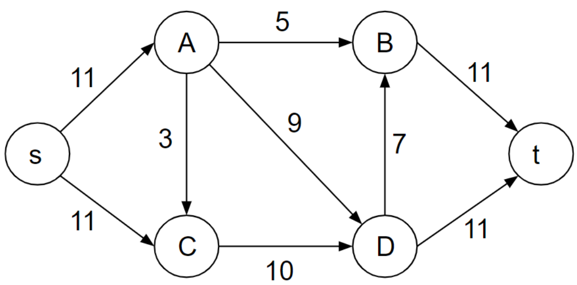

Figure 1: A flow network.

Good Luck!

Step	Augmenting path	Residual capacity	Flow value
1	$s \to A \to D \to t$		
2	$s \to C \to D \to t$		
3	$s \to C \to D \to A \to B \to t$		
4	$s \to A \to D \to B \to t$		
5			

Table 1: Augmenting paths to be used within the Ford-Fulkerson method.