Parameterized algorithms. (5 and 12 of April 2021.)

Relevant material can be found in Chapter 2 of the BWCA book (references in questions refer to that book), and material on treewidth can be found in selected parts of lecture notes posted of my home page.

Homework assignment (hand in by April 26):

1. (Exercise 2.2, page 50.) In the *Cluster Editing* problem, we are given a graph G and an integer k, and the objective is to check whether we can turn G into a cluster graph (a disjoint union of cliques) by making at most k edge editions, where each edition is adding or deleting one edge. Obtain a $3^k n^{O(1)}$-time algorithm for *Cluster Editing*.

2. (Exercise 2.3, page 51.) Recall the *Set Splitting* problem (Section 2.2.1), and suppose that the input instance is such that there is an assignment that splits k sets. Prove that a random assignment splits k sets with probability at least 2^{-k}. (In class we only showed that this probability is at least 2^{-2k}.)

3. In the *Disjoint Paths* problem the input is an undirected graph G and k pairs of vertices $(s_1, t_1), \ldots, (s_k, t_k)$. The goal is to find k vertex disjoint paths $P_1, \ldots P_k$ (no two paths share a vertex), where for every $1 \le i \le k$, path P_i connects s_i with t_i. A major result of Robertson of Seymour shows that the problem is fixed parameter tractable, with k as a parameter. Hence for every fixed k the problem can be solved in polynomial time. (For directed graphs, the problem is NP-hard already for $k = 2$.) Their algorithm laid much of the groundwork for the theory of parameterized algorithms. It is composed of several components. In one component one assumes that G has treewidth bounded by a parameter p (that depends on k). Design an algorithm (and prove its correctness) that when $p > k$, solves the *Disjoint Paths* problem in time $f(p)n^{O(1)}$ (for some function f of your choice). You may assume that a tree decomposition of G with width p is given to your algorithm.