Practical session 1

Alert: Read the Instructions in the README file on Madoc.

Write methods for the tasks below. For each of them, provide in a comment its running time (worst case). When the question does not indicate the data structure(s) to use, you must choose one the structure that provides the best worst case running time[1].

Exercise 1. Given a directed graph G, compute the *transposed* graph G^t that has the same vertices as G but all the arcs are reversed. Two parameters must allow the user to choose the representation (adjacency list or adjacency matrix) of the input graph G, and of the output graph G^t (which means that you should write four variants of the method).

Exercise 2. Given an undirected graph G and a sequence of vertices $v_1, v_2, \ldots, v_k$, test whether there is a path in G with edges v_1v_2, $v_2v_3, \ldots, v_iv_{i+1}, \ldots, v_{k-1}v_k$. A parameter must allow to choose the representation (adjacency list or adjacency matrix) of the input graph G.

[1] Recall that hashsets and hashmaps are NOT constant from the worst case point of view, but linear; JDK8 and next combine these structures with red-black trees in order to achieve $\theta(logn)$ in the worst case, but we are not working with JDK8