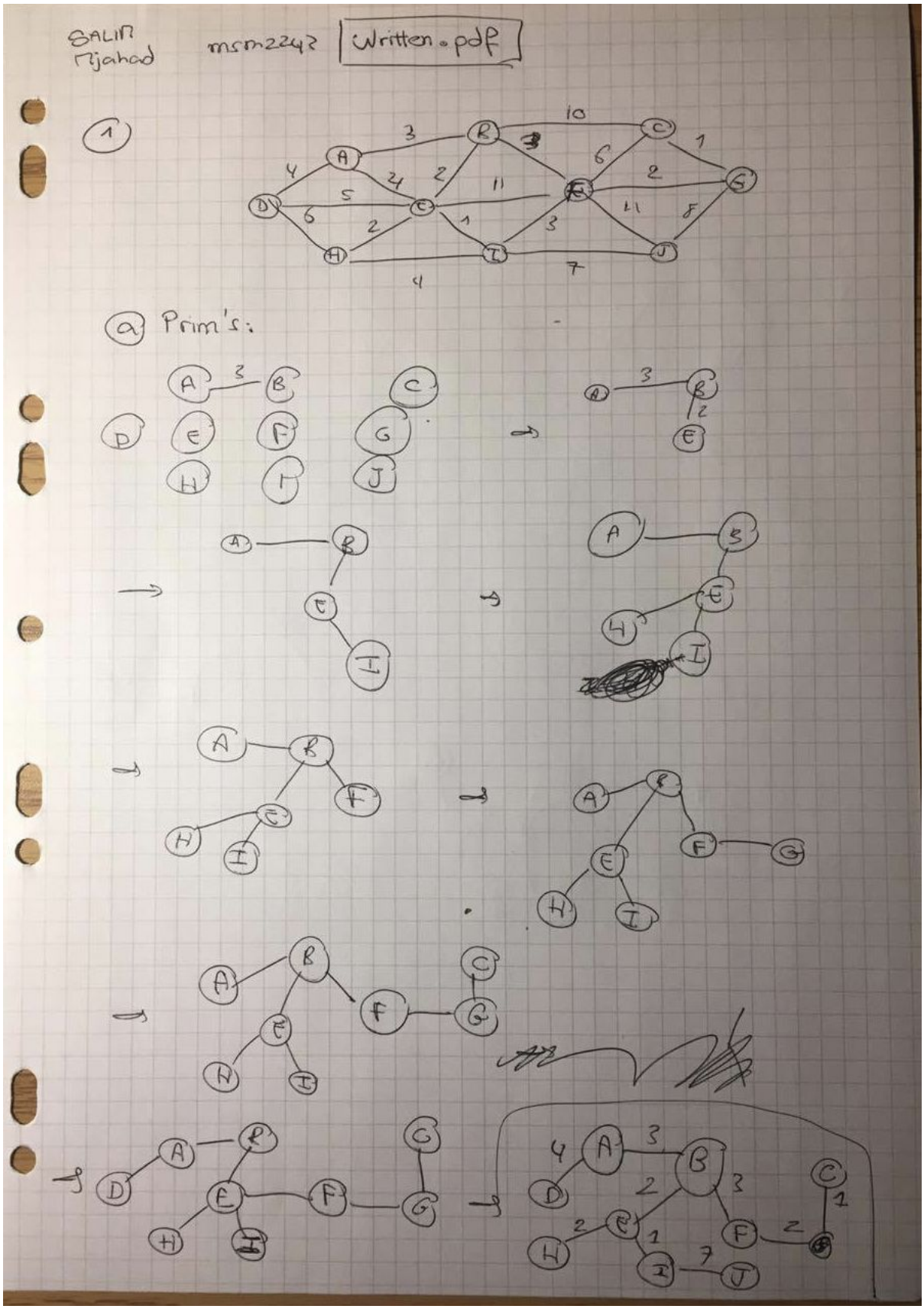
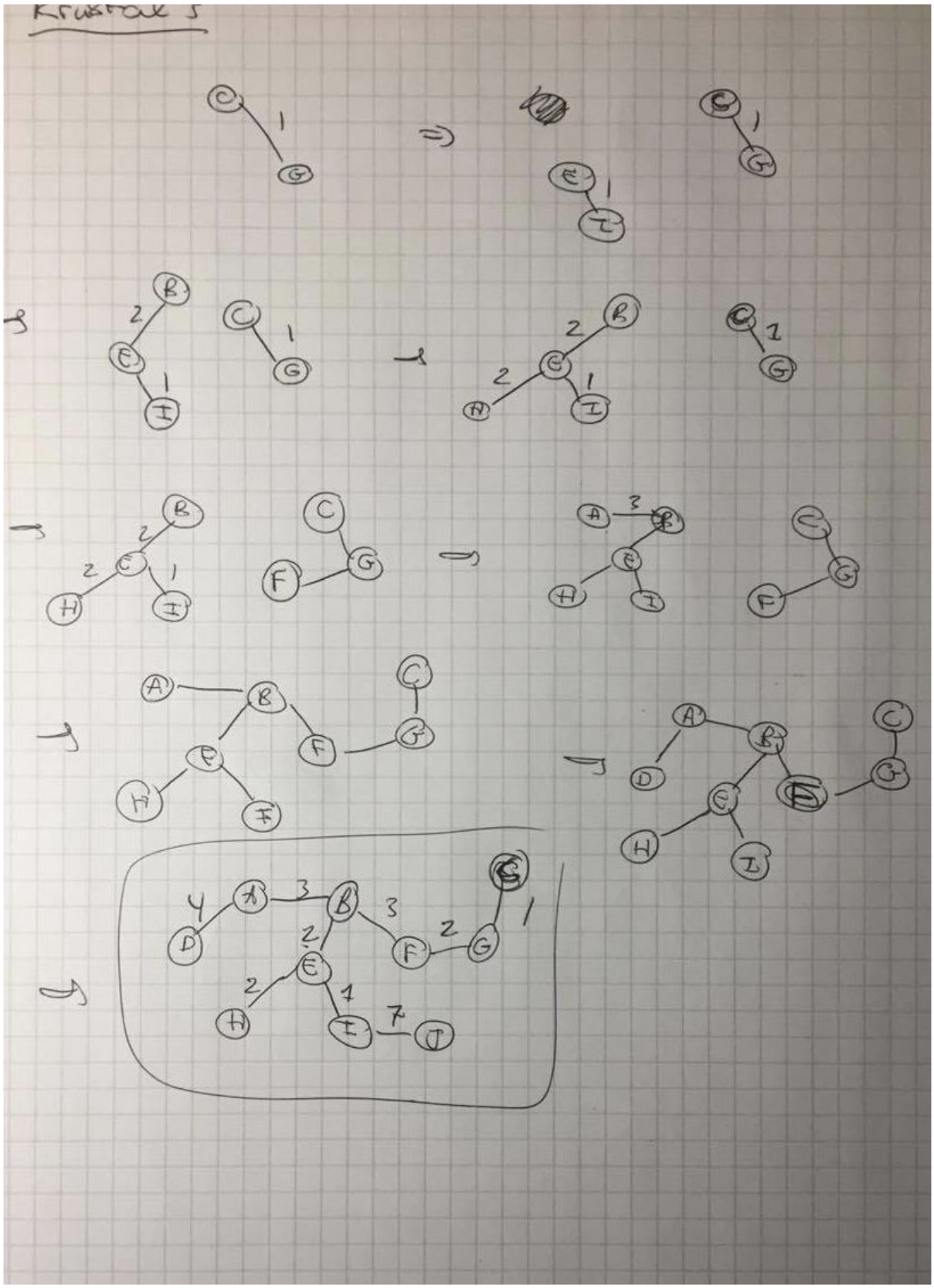
P1



Krukal 5



P3

a) We can construct a graph such that the vertices represent the actors and the edges between them mean they played in a same movie. The problem then becomes that of finding the shortest path from Bacon to the actor in question. So the bacon number of the actor is the length of the path. Finding that shortest path is easy. An algorithm BFS in the book (section 9.3.1) explains how it works. We iterate starting from Kevin Bacon then change the value of distance stored in each vertex that is unknown and adjacent to the current vector. Since the edges are unweighted, every update of distance is an addition of +1. We keep checking until we reach the vector which represent our actor.

b) This problem is even easier than the above. After running the algorithm a, we have with each vertex a number representing their Bacon number, we then only have to iterate through those vertices while keeping track of the highest bacon number (BNMax) and the actor with that number (MaxActor). We just update them whenever we find a higher number.

c) This is very similar to problem in a. Now, we just need to change the starting point of the algorithm from Kevin Bacon to whichever actor we want to check then run the same algorithm.