1. This question is about Markov Models (MM). We consider a MM on a finite set of states S that generates a sequence of letters from a finite alphabet Σ. The parameters of the model are the initial probabilities $(p(s))_{s\in S}$, the transition matrix $(m_{st})_{s,t\in S}$, and the emission probabilities $(e_s(a))_{s\in S,a\in\Sigma}$.

   (a) We first assume that that for each state $s \in S$, $\sum_{a\in\Sigma} e_s(a) \leq 1$, and that with probability $1 - \sum_{a\in\Sigma} e_s(a)$ the state s stays silent, i.e. doesn't generate a letter. Design an algorithm that takes a sequence of n letters as an input and outputs the most likely sequence of states that generated the input. Prove the correctness of your algorithm and estimate the running time and the space used in terms of n and the number of states of the model k. **[25 marks]**

   (b) We now return to the more traditional setup, in which $\sum_{a\in\Sigma} e_s(a) = 1$ for each state $s \in S$. Consider the question of estimating the parameters of the model, given a sequence of observed letters (but not the sequence of states that generated it) and assuming that the number of states k is also given. Implement the Expectation-Maximisation (EM) algorithm for this problem. The implementation should be in a single Python file. There should be instructions at the beginning detailing how I can run it on an input of my choice and what the output format is. **[15 marks]** Write a brief report that first gives a self-contained description of the EM algorithm, and then provides a validation of your implementation. **[15 marks]**

2. This question is about tree reconstruction, and in particular about the BUILD algorithm from [1].

   (a) Explain what the algorithm does and how it works in your own words. Do not use pseudocode. **[10 marks]**

   (b) Expand the partition step (given below) in pseudocode

   $$\text{compute } \pi_C = S_1, S_2, \ldots S_r;$$

   [10 marks]

   (c) Run the algorithm on the following set of constraints

   $$\begin{array}{ll} (e,f) < (k,d) & (c,l) < (g,k) \\ (c,h) < (a,n) & (g,b) < (g,i) \\ (j,n) < (j,l) & (g,i) < (d,m) \\ (c,a) < (f,h) & (c,h) < (c,a) \\ (j,l) < (e,n) & (e,f) < (h,l) \\ (n,l) < (a,f) & (j,l) < (j,a) \\ (d,i) < (k,n) & (k,m) < (e,i) \\ (d,i) < (g,i) & (j,n) < (j,f)\,. \end{array}$$

   You should show the partitioning and the recursive calls at each stage. **[10 marks]**

   (d) "Reverse" the BUILD algorithm, i.e. design an algorithm that takes a tree with labeled leaves as an input, and produces a set of constraints of the form $(i,j) < (k,l)$, such that when BUILD runs on that set, the result is (an isomorphic copy of) the input tree. Prove the correctness of your algorithm. Also, a smaller output (a set of constraints) would give you a better mark. **[15 marks]**

References

[1] A. V. Aho, Y. Sagiv, T. G. Szymanski, J. D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. SIAM Journal on Computing, 10:405—421, 1981.