Bioinformatics assignment
L3 students should **ignore** the questions in **boldface** (for a total of 100). L4 students should complete all questions (for a total of 150).

1. This question is about the BUILD algorithm from [1].

   (a) Explain what the algorithm does and how it works in your own words. Do not use pseudocode [25 marks]

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

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

   [20 marks]

   (c) **Write a recurrence that expresses the running time of BUILD depending on the number of different leaf-labels n and the number of constraints m. Use it to estimate the running time of the algorithm assuming that the partitioning step runs in time $f(n, m)$ for some function monotonically nondecreasing function f. [25 marks]**

   (d) 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. [25 marks]

   (e) "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 $(i, j) < (k, l)$, which is consistent only with (an isomorphic copy of) that input. Prove the correctness of your algorithm. Also, a smaller output (a set of constraints) would give you a better mark. [30 marks]

2. **This question is about the MINCUTSUPERTREE algorithm from [2].**

   (a) **One of its properties is that it preserves nesting and subtrees that are shared by all of the input trees. Point where precisely in the algorithm this property is achieved. [15 marks]**

   (b) **Argue that the MINCUTSUPERTREE algorithm is a generalisation of the BUILD algorithm, i.e. show how to encode a constraint from the inputs of the later as a tree, which is one of the inputs of the former. [10 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.

[2] C. Semple and M. Steel. A supertree method for rooted trees. Discrete Applied Mathematics, 105:147-158, 2000.