

CS583 -- Programming Assignment -- Fall 2019 -- due Friday, Nov. 20

This concerns the rotation distance between trees. We will use a paper, "Complete Balancing via Rotation" (1916), available on-line:

web.science.mq.edu.au/~bmans/BMans/Papers/LMMP_CJ2016.pdf

In particular sections 1-4. You will code three algorithms: A_1 , A_2 and A_3 . You will verify the claims in Theorems 1, 2 and 3 empirically. (It is possible the theorems will be off by a constant amount, there are a few errors on the paper.) You will do this for 5 trees each of size $n=1000, 1100, 1200$.

There is a video explaining parts of the paper that I feel deserve further comment.

You will start each experiment with a tree S created from the fixed tree T , for those n . You will take that initial T and mark each edge randomly as "do not rotate" with probability 0.01. Then you will rotate each remaining edge with probability 0.5. (These constants may be changed if they do not illustrate the Theorems.) After S is created, of course, the algorithms will be unaware of how it was created.

You will submit your (executable) code written in C++, Python or Java (otherwise ask) to the TA at arana6@gmu.edu. You will also submit a text document. It will include a short (~one-page) write-up of your findings. Further you will include any instructions the TA will need to test your code. You should mention any decisions you made when you felt the assignment was ambiguous.