WARSAW UNIVERSITY OF TECHNOLOGY

Faculty of Mathematics and Information Science

ADVANCED ALGORITHMS

Final

Documentation IS

in Trees

Created by:
Dmytro Narepekha, Jianhao Luo

1. List of modifications

- No modifications were made.

2. User Manual

2.1. Prepare input file

First of all it is mandatory to verify that the device (laptop or PC), that the user are using to run the app, has PYTHON installed on it, including numpy.

In case python is not install, go to https://www.python.org/downloads/ and download it, after that install it and be sure to check *(add python to PATH)*.
After finishing installing python, open your command prompt and write "*pip install numpy*" to download numpy.

Generate Trees by running: python inputFileGenerator.py
Run algorithm by: python main.py
where you pick the mode 1. Test Mode, no output.txt, but performance is measured 2. Output mode, where IS Count is written to output.txt, no measurement is done.

2.2. Algorithm execution

After creating the input file, in order to execute the algorithm and calculate the IS count in tree, it is required to run main.*py*. That should be done using command line:

-Firstly, be sure that you need to write the command in the same directory that contains the main.py file, if not use cd (for example: cd C:\Users\dm\Desktop\ISinTrees\main.py).

-Secondly the result of the the programm execution will be seen in the output file output.txt.

3. Test

3.1. Measurement (Time Complexity Test) and approach to the test

We utilized timeit library, which measure the algorithm by executing N times and take the average execution time. In our case, we pick the N = 1000.

Tree is generated as follow: each node has random child number from 1 to 3 (inclusive).

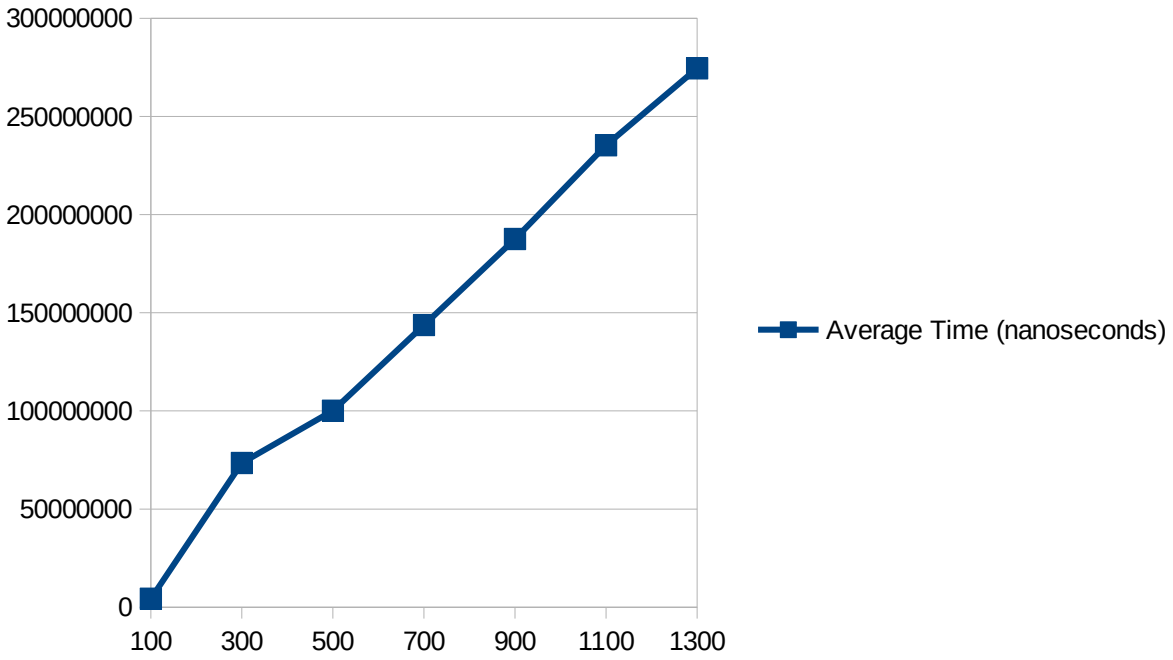

Figure 1: Peformance(Nano second) VS Total Node Number

Number of measuring iterations	***Total number of nodes***	***Average Time (nanoseconds)***
1000	***100***	***4315383***
1000	***300***	***73472232***
1000	***500***	***100095212***
1000	***700***	***143735182***
1000	***900***	***187550760***
1000	***1100***	***235331145***
1000	***1300***	***274462781***

As the number increases, the trend is linear-like.

3.2. Correctness Test

Number of nodes	*IS Count*
5	***14***
6	***22***
7	***36***
8	***69***

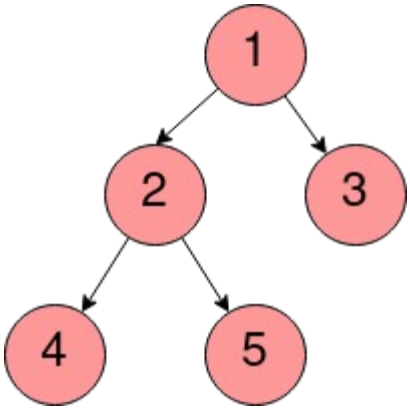

Figure 4: Node Number 5

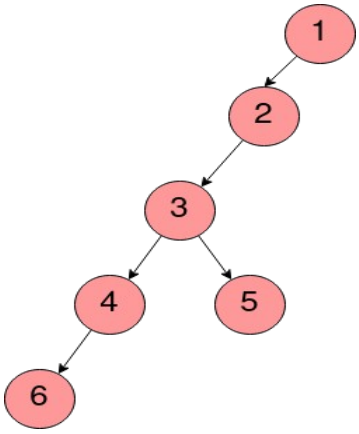

Figure 2: Node Number: 6

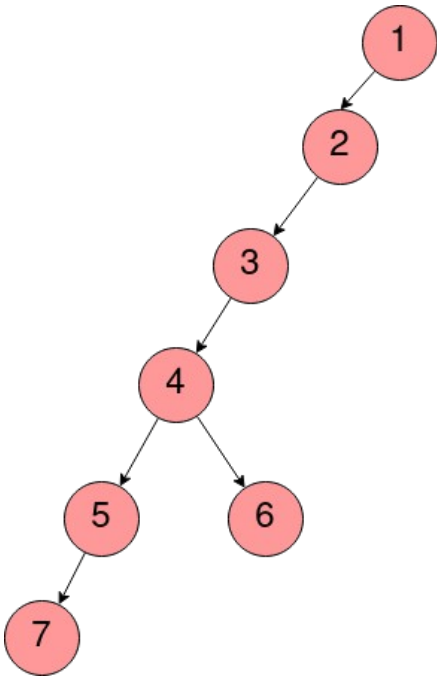

Figure 3: Node Number: 7

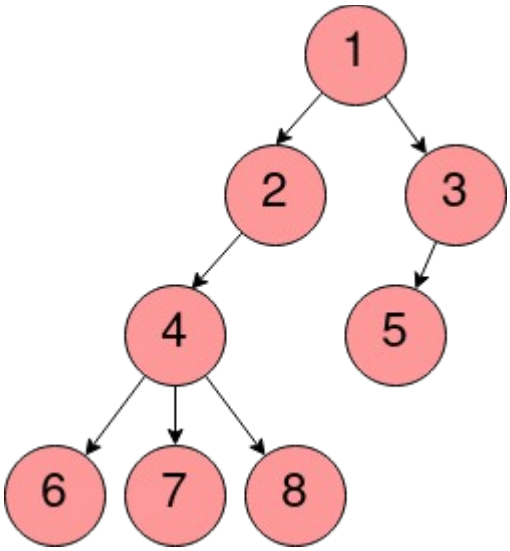

Figure 5: Node Number: 8

4. Job partition

#	Tasks	Jianhao Luo	Dmytro Narepekha
1	Solution Description	X	X
2	Pseudocode Description	X	
3	Correctness and Time Complexity Analysis	X	
4	Testing and Performance Analysis		X
5	Test and Input File Generator		X
6	Algorithm Development and Implementation	X	X
7	Final Report	X	X