2XC3 Lab 4-5

By: Abdullah Khan, Abdul-Hadi Siddiqui, Ziyu Li

|  |  |
| --- | --- |
| TABLE OF CONTENTS | Pgs. |
| Experiment 1 |  |
| Experiment 2 |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Appendix | pg -pg 32 |

Experiment 1

Outline

The purpose of this experiment is to see how often a graph contains cycles. We know that if a graph has *i* nodes then its spanning tree will contain *i-1* edges. There for, we know that if a graph as more than *i-1* edges it will contain a cycle. For this experiment we wanted to see the probability that a graph contains a cycle given that it has less than *i* edges. To do this, we tested the graph of size 100, 90 and 80, we had their number of edges range from 0-99, 0-89, 0-79 respectively. We had an epoch value of 1000 for the probability (see Appendix “Experiment 1” sub section “Epoch” too see how we calculate probabilities).

Results

Chart, line chart

Description automatically generated

Discussion

As we can see the probability increases more we increase the total number of edges. These probabilities approach one as the number of edges gets closer to the number nodes. This makes sense as if there are more than *v-1* edges in a graph with *v* nodes, it will contain cycles. Its also important to note that the probabilities do not grow as quickly when the size of the graphs are bigger. The bigger the total number of nodes there are the smaller the chances that a certain number of edges will from a cycle since there are more ways for them not to from a cycle.

Appendix

We have figure number on top /bottom of each figure, and we also mentioned figure number that we are generating in the python. To be clearer, we listed the python code line number so that it’s easier to track.

Experiment 1

Epoch:

To calculate the probability of graph a with x nodes and y randomly generated edges, containing a cycle we need to run a certain number of tests. The Epoch value is the total number of times we run the test. We have a counter which starts at zero and increments by one each time a graph contains a cycle. After the tests are done we calculate the probability by doing quotient between the number of times we found a graph with a cycle and the Epoch value which was the total number of times we ran the tests.

Figure 1.1: experiment1.py line 95-153

Figure 1.2: experiment1.py line 155-213

Figure 1.3: experiment1.py line 215-273

Figure 1.4: experiment1.py line 275-337

Experiment 2

Figure 2.1: experiment2.py line 318-325

Figure 2.2: experiment2.py line 327-334

Figure 2.3: experiment2.py line 336-343

Figure 2.4: experiment2.py line 345-352

Figure 2.5: experiment2.py line 357-364

Figure 2.6: experiment2.py line 366-373

Figure 2.7: experiment2.py line 375-382

Figure 2.8: experiment2.py line 385-392

Experiment 3

Scale:

We get the data and graph by using a for loop which ranges from 0 to the max number of swaps. The Scaling factor is simply the step size we take in our range function. This has the effect of smoothing out the graph by reducing the variance and reducing the amount of time it takes to calculate the entire data set.

Figure 3.1: Exp3.py line 147

Experiment 4

Figure 4.1: Exp4.py line 334

Figure 4.2: Exp4.py line 338

Figure 4.3: Exp4.py line 341

Figure 4.4: Exp4.py line 344

Experiment 5

Figure 5.1: Exp5.py line 218

Experiment 6

Figure 6.1: experiment6.py line 197-204

Figure 6.2: experiment6.py line 207-214

Figure 6.3: experiment6.py line 217-224

Figure 6.4: experiment6.py line 227-234

Experiment 7

Figure 7.1: experiment7topBottom.py line 127-133

Figure 7.2: experiment7bottomUp.py line 209.215

Figure 7.3: experiment7topBottom.py line 137-143

Figure 7.4: experiment7bottomTop.py line 219.225

Figure 7.5: experiment7topBottom.py line 147-153

Figure 7.6: experiment7bottomTop.py line 229-235

Figure 7.7: experiment7topBottom.py line 157-163

Figure 7.8: experiment7bottomTop.py line 239-245

Experiment 8

Figure 8.1: experiment8.py line 267-275

Figure 8.2: experiment8.py line 279-287

Figure 8.3: experiment8.py line 290-298

Figure 8.4: experiment8.py line 301-309

Figure 8.5: experiment8.py line 313-321

Figure 8.6: experiment8.py line 325-333

Figure 8.7: experiment8.py line 337-345

Figure 8.8: experiment8.py line 348-356