Kyle Johnson

Professor Skaggs

ETEC 3401 9:00 AM

October 18, 2020

**Calculating Derangements Given a Number Permutations**

Based on the python program that I made that would calculate the runtimes and correctness of the different formulas we derived during class, I was able to format a table of my results. Each of the following methods come with their own pros and cons.

The function that was made from the pattern that some of the students found in class resulted in a pattern that had a fast runtime but with a greater number of permutations. It would become inaccurate. The guessing function was also able to be solved using recursion. The average runtime based on a number of trials up to 43 elements was 0.00249834 seconds.

The function that was based on inclusion and exclusion resulted in a function that was derived using calculus. This method resulted in a function that was able to use recursion as well as a proof using induction. This method resulted in the fastest runtime while also giving the most accurate results when compared to the guessing example which was the only other function comparable in runtime. The average runtime based on a number of trials up to 43 elements was 0.002493284 seconds.

The next function was based on the secret santa scenario. This function was accurate but as the number of permutations gets higher and higher the runtime gets slower and slower but the accuracy for the number of derangements is still good. This method is able to also take advantage of recursion. The average runtime based on a number of trials up to 43 elements was 9.886983579 seconds.

Finally, I did the function based on counting. This method is accurate because it’s making use of computer power but if it was done by hand it’s likely to have human error and take even longer. Although, even making use of computing power it still fails after about 12 elements due to a memory leak. This method is unable to take advantage of recursion as well. The average runtime based on a number of trials until the memory leak at 13 elements was 74.87105315 seconds.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Time Elapsed (Seconds)** | | | |
| **Number of Elements Permutations** | **Based on Guessing** | **Based on Inclusion and Exclusion** | **Based on given 2 possibilities of a value appearing (secret santa)** | **Based on Counting** |
| 0 | 0.002991676 | 0.002990723 | 0.002993107 | 0.002992153 |
| 1 | 0.001993179 | 0.001994371 | 0.000997782 | 0.000997543 |
| 2 | 0.001993418 | 0.000997543 | 0.000997305 | 0.000997066 |
| 3 | 0.000997066 | 0.001996994 | 0.002991676 | 0.001993179 |
| 4 | 0.000997782 | 0.000997066 | 0.001993418 | 0.002991676 |
| 5 | 0.002992153 | 0.001993895 | 0.001995087 | 0.00199461 |
| 6 | 0.002991438 | 0.002992153 | 0.002991915 | 0.002991199 |
| 7 | 0.001994848 | 0.002992868 | 0.002992153 | 0.006982327 |
| 8 | 0.002991676 | 0.002991915 | 0.002991199 | 0.042885065 |
| 9 | 0.00199461 | 0.001994848 | 0.002994299 | 0.507641792 |
| 10 | 0.003987789 | 0.002992153 | 0.002992153 | 5.105942011 |
| 11 | 0.001995087 | 0.003988028 | 0.001994371 | 59.66232777 |
| 12 | 0.002993107 | 0.002992153 | 0.002992153 | 907.9829545 |
| 13 | 0.003989935 | 0.002991199 | 0.002992153 | ERROR: MEMORY LEAK |
| 14 | 0.00199461 | 0.00199461 | 0.001994371 | ERROR: MEMORY LEAK |
| 15 | 0.002214193 | 0.000997066 | 0.001994848 | ERROR: MEMORY LEAK |
| 16 | 0.000997782 | 0.000997305 | 0.001995802 | ERROR: MEMORY LEAK |
| 17 | 0.000997543 | 0.000996828 | 0.001994371 | ERROR: MEMORY LEAK |
| 18 | 0.002991438 | 0.00199461 | 0.002992153 | ERROR: MEMORY LEAK |
| 19 | 0.002991676 | 0.002992153 | 0.00399065 | ERROR: MEMORY LEAK |
| 20 | 0.00199461 | 0.002991915 | 0.005981207 | ERROR: MEMORY LEAK |
| 21 | 0.00199461 | 0.00199461 | 0.00698185 | ERROR: MEMORY LEAK |
| 22 | 0.002991915 | 0.001994371 | 0.008482218 | ERROR: MEMORY LEAK |
| 23 | 0.002992153 | 0.002992153 | 0.014960051 | ERROR: MEMORY LEAK |
| 24 | 0.001994848 | 0.001994848 | 0.020241737 | ERROR: MEMORY LEAK |
| 25 | 0.00199461 | 0.002990961 | 0.02992034 | ERROR: MEMORY LEAK |
| 26 | 0.002992153 | 0.00299263 | 0.045876741 | ERROR: MEMORY LEAK |
| 27 | 0.00199461 | 0.001994371 | 0.104134321 | ERROR: MEMORY LEAK |
| 28 | 0.00199461 | 0.001994848 | 0.118681431 | ERROR: MEMORY LEAK |
| 29 | 0.00199461 | 0.002992392 | 0.205448627 | ERROR: MEMORY LEAK |
| 30 | 0.002991915 | 0.002991199 | 0.306181192 | ERROR: MEMORY LEAK |
| 31 | 0.003990412 | 0.002992392 | 0.521605492 | ERROR: MEMORY LEAK |
| 32 | 0.002992153 | 0.002991676 | 0.813349485 | ERROR: MEMORY LEAK |
| 33 | 0.002992153 | 0.002992153 | 1.322492361 | ERROR: MEMORY LEAK |
| 34 | 0.002992392 | 0.002992392 | 2.034585476 | ERROR: MEMORY LEAK |
| 35 | 0.001994848 | 0.001994848 | 3.352083206 | ERROR: MEMORY LEAK |
| 36 | 0.001994371 | 0.00199461 | 5.590605736 | ERROR: MEMORY LEAK |
| 37 | 0.003989935 | 0.00199461 | 8.940227985 | ERROR: MEMORY LEAK |
| 38 | 0.001995087 | 0.00199461 | 14.4186511 | ERROR: MEMORY LEAK |
| 39 | 0.002992392 | 0.002991915 | 23.35685802 | ERROR: MEMORY LEAK |
| 40 | 0.00199461 | 0.002990961 | 38.16353774 | ERROR: MEMORY LEAK |
| 41 | 0.002992153 | 0.00299263 | 64.55835366 | ERROR: MEMORY LEAK |
| 42 | 0.002992153 | 0.002992392 | 105.2219388 | ERROR: MEMORY LEAK |
| 43 | 0.00299263 | 0.004986525 | 165.8162277 | ERROR: MEMORY LEAK |
| Averages: | 0.00249834 | 0.002493284 | 9.886983579 | 74.87105315 |