*Jorge Cruz*

*CS370 | [Company Address]*

*graphs & analysis*

Two Dimensions

|  |  |  |
| --- | --- | --- |
| **n** | ****Rn2 | **F**saw(n) |
| 10 | 26.245369 | 0.04206423 |
| 11 | 30.019513 | 0.02867326 |
| 12 | 34.1607608 | 0.01937201 |
| 13 | 38.2572428 | 0.01313242 |
| 14 | 42.7928462 | 0.00886356 |
| 15 | 47.2450309 | 0.005979 |
| 16 | 51.9983617 | 0.00401626 |
| 17 | 56.7461398 | 0.00269739 |
| 18 | 61.7517534 | 0.00182077 |
| 19 | 66.8455806 | 0.0012185 |
| 20 | 72.042113 | 8.19E-04 |
| 21 | 77.4183969 | 5.49E-04 |
| 22 | 82.6173395 | 3.69E-04 |
| 23 | 88.8974233 | 2.45E-04 |
| 24 | 93.7252159 | 1.64E-04 |
| 25 | 99.6471763 | 1.10E-04 |
| 26 | 105.881004 | 7.33E-05 |
| 27 | 112.080903 | 4.87E-05 |
| 28 | 118.671045 | 3.35E-05 |
| 29 | 121.907236 | 2.16E-05 |
| 30 | 134.813652 | 1.47E-05 |
| 31 | 135.010215 | 9.79E-06 |
| 32 | 145.584496 | 6.45E-06 |
| 33 | 152.025522 | 4.31E-06 |
| 34 | 151.480702 | 2.85E-06 |
| 35 | 181.76555 | 2.09E-06 |
| 36 | 169.045455 | 1.32E-06 |
| 37 | 140.2 | 8.50E-07 |
| 38 | 203.37931 | 5.80E-07 |
| 39 | 201 | 3.20E-07 |
| 40 | 221.096774 | 3.10E-07 |

α2=1.4756, β2=0.401

Software and design Optimizations

Software and design Optimizations

* HashMap
* Integrated the use of HashMap container to search if set of points (x, y) already lie on the grid. For every step, the program would go through an iteration of a loop of n steps, it would pass a new int array of size dimension into a function that turns it into a string (next points on graph) and checks if that string is in HashMap. If it is in HashMap then program would break out of loop and go through next iteration. Checking the HashMap is in O(n) order, so it saves time compared to other containers.
* Polymorphism
* Integrated the use of polymorphism for many reasons. One there was a few similar methods within the children class which I can call directly from the parent object. Another is because, the program uses many instanceOf comparisons compared to a traditional == comparison. The instanceof method provides better benchmark times. Using parent class also makes it more readable to user.
* Another integration to optimize my program was the inclusion of adding both n\_saw and square distance total inside the for loops instead of pulling them out and adding another loop to add these totals. I use this whenever possible to exclude as many loops as possible as loops are O(n) costly.

Higher Dimensions

Walks in 3rd dimension

|  |  |  |
| --- | --- | --- |
| **n** | 〈Rn2〉 **** | **F**saw(n) |
| 10 | 16.7965414 | 0.1456901 |
| 11 | 18.8317412 | 0.1155862 |
| 12 | 20.9649441 | 0.0912486 |
| 13 | 23.0588446 | 0.0722955 |
| 14 | 25.2084273 | 0.0570007 |
| 15 | 27.3795083 | 0.0451305 |
| 16 | 29.6147 | 0.0355292 |
| 17 | 31.9040342 | 0.0280204 |
| 18 | 34.1122791 | 0.0219364 |
| 19 | 36.4705927 | 0.0172491 |
| 20 | 38.8005612 | 0.0136844 |
| 21 | 41.1497592 | 0.0107359 |
| 22 | 43.5469531 | 0.0084265 |
| 23 | 45.9559826 | 0.0066133 |
| 24 | 48.1289976 | 0.0052156 |
| 25 | 50.7566084 | 0.004116 |
| 26 | 52.8870724 | 0.0032419 |
| 27 | 55.2741037 | 0.0025494 |
| 28 | 57.8620342 | 0.001986 |
| 29 | 60.4392027 | 0.0015453 |
| 30 | 63.009986 | 0.0012117 |
| 31 | 65.9781037 | 9.55E-04 |
| 32 | 68.3773434 | 7.52E-04 |
| 33 | 71.7994661 | 5.99E-04 |
| 34 | 73.2075232 | 4.73E-04 |
| 35 | 76.4648603 | 3.54E-04 |
| 36 | 79.363156 | 2.84E-04 |
| 37 | 82.3428571 | 2.24E-04 |
| 38 | 84.1319797 | 1.77E-04 |
| 39 | 86.807386 | 1.38E-04 |
| 40 | 86.3367253 | 1.08E-04 |

α3=1.2043, β3=0.241

|  |  |  |
| --- | --- | --- |
| **n** | ****Rn2 | **F**saw(n) |
| 10 | 14.02087612 | 0.2578065 |
| 11 | 15.56281222 | 0.2194963 |
| 12 | 17.16192863 | 0.1869657 |
| 13 | 18.72240305 | 0.158898 |
| 14 | 20.32261531 | 0.1351089 |
| 15 | 21.92238714 | 0.1149294 |
| 16 | 23.53405558 | 0.0977593 |
| 17 | 25.13069593 | 0.0831625 |
| 18 | 26.82117241 | 0.070658 |
| 19 | 28.42837562 | 0.0599705 |
| 20 | 30.08596965 | 0.0509645 |
| 21 | 31.69814798 | 0.043196 |
| 22 | 33.44466319 | 0.0367748 |
| 23 | 34.97593617 | 0.0312336 |
| 24 | 36.63344322 | 0.0264813 |
| 25 | 38.34615899 | 0.0224394 |
| 26 | 40.0338855 | 0.0190583 |
| 27 | 41.74785734 | 0.0161365 |
| 28 | 43.3642359 | 0.0137227 |
| 29 | 45.06133625 | 0.0115951 |
| 30 | 46.64850854 | 0.0098762 |
| 31 | 48.45607425 | 0.0083607 |
| 32 | 50.25388266 | 0.0071214 |
| 33 | 51.83254425 | 0.0060619 |
| 34 | 53.42782891 | 0.0051177 |
| 35 | 55.39819068 | 0.004311 |
| 36 | 56.86351218 | 0.0036604 |
| 37 | 58.39040191 | 0.0031027 |
| 38 | 60.02698526 | 0.0026533 |
| 39 | 62.54367816 | 0.002262 |
| 40 | 63.82422928 | 0.0018911 |

Walks in 4th dimension

α4=1.0921, β4=0.164

Self-avoiding polygons

Self-avoiding polygons

γ2=0.489;

Polygon in 2nd dimension

|  |  |
| --- | --- |
| **n** | FSAP(n) |
| 10 | 5.13E-04 |
| 12 | 1.73E-04 |
| 14 | 6.02E-05 |
| 16 | 2.26E-05 |
| 18 | 8.36E-06 |
| 20 | 3.12E-06 |
| 22 | 1.10E-06 |
| 24 | 4.60E-07 |
| 26 | 1.50E-07 |
| 28 | 1.10E-07 |
| 30 | 2.00E-08 |
| 32 | 1.00E-08 |
| 33 | 0 |
| 34 | 0 |
| 35 | 0 |
| 36 | 0 |
| 37 | 0 |
| 38 | 0 |
| 39 | 0 |
| 40 | 0 |

γ3=0.324;

Polygon in 3rd dimension

|  |  |
| --- | --- |
| **n** | FSAP(n) |
| 10 | 7.20E-04 |
| 12 | 3.29E-04 |
| 14 | 1.57E-04 |
| 16 | 7.56E-05 |
| 18 | 3.66E-05 |
| 20 | 1.88E-05 |
| 22 | 1.04E-05 |
| 24 | 5.06E-06 |
| 26 | 2.87E-06 |
| 28 | 1.41E-06 |
| 30 | 8.70E-07 |
| 32 | 4.00E-07 |
| 34 | 2.40E-07 |
| 36 | 1.10E-07 |
| 38 | 8.00E-08 |
| 40 | 4.00E-08 |

γ4=0.263;

Polygon in 4th dimension

|  |  |
| --- | --- |
| **n** | FSAP(n) |
| 10 | 5.32E-04 |
| 12 | 2.65E-04 |
| 14 | 1.42E-04 |
| 16 | 7.72E-05 |
| 18 | 4.40E-05 |
| 20 | 2.51E-05 |
| 22 | 1.45E-05 |
| 24 | 9.21E-06 |
| 26 | 5.42E-06 |
| 28 | 3.34E-06 |
| 30 | 1.89E-06 |
| 32 | 1.24E-06 |
| 34 | 7.50E-07 |
| 36 | 4.80E-07 |
| 38 | 3.10E-07 |
| 40 | 1.50E-07 |