Nick Petty

NTH FIBONACCI NUMBER

1. DESCRIPTION

The Fibonacci numbers are a sequence of integers defined by the recurrence relation $F_n = F_{n-1} + F_{n-2}$, where F_n is the n^{th} number in the series and $F_0 = 0$ and $F_1 = 1$. This sequence appears frequently in mathematics, computer science, and even biology, and has been described in mathematical texts for centuries.

2. APPLICATIONS

A. Golden Ratio – F_n / F_{n-1} approaches φ as n approaches ∞

B. Fibonacci heap – data structure for priority queues

C. Hilbert's Tenth Problem – Fibonacci numbers used to show unsolvability

D. Bee ancestry – bee reproduction creates an unusual number of ancestors

E. Brock-Mirman model – a generalized sequence is used in an optimal control function

F. Fibonacci Quarterly & the Fibonacci Association – publishing scholarly work since 1963

3. COMPETING ALGORITHMS

A. Recursion

i. Directly implement recurrence relation $F_n = F_{n-1} + F_{n-2}$, base cases $F_0 = 0$ and $F_1 = 1$

ii. Creates a recursion tree of height n where each level, L, has at most 2^L sub problems

iii. $T(n) = T(n-1) + T(n-2) \rightarrow O(2^n)$

B. Dynamic Programming

i. Store the previously calculated F_{n-1}, F_{n-2} in an array, starting with $F_0 = 0$ and $F_1 = 1$

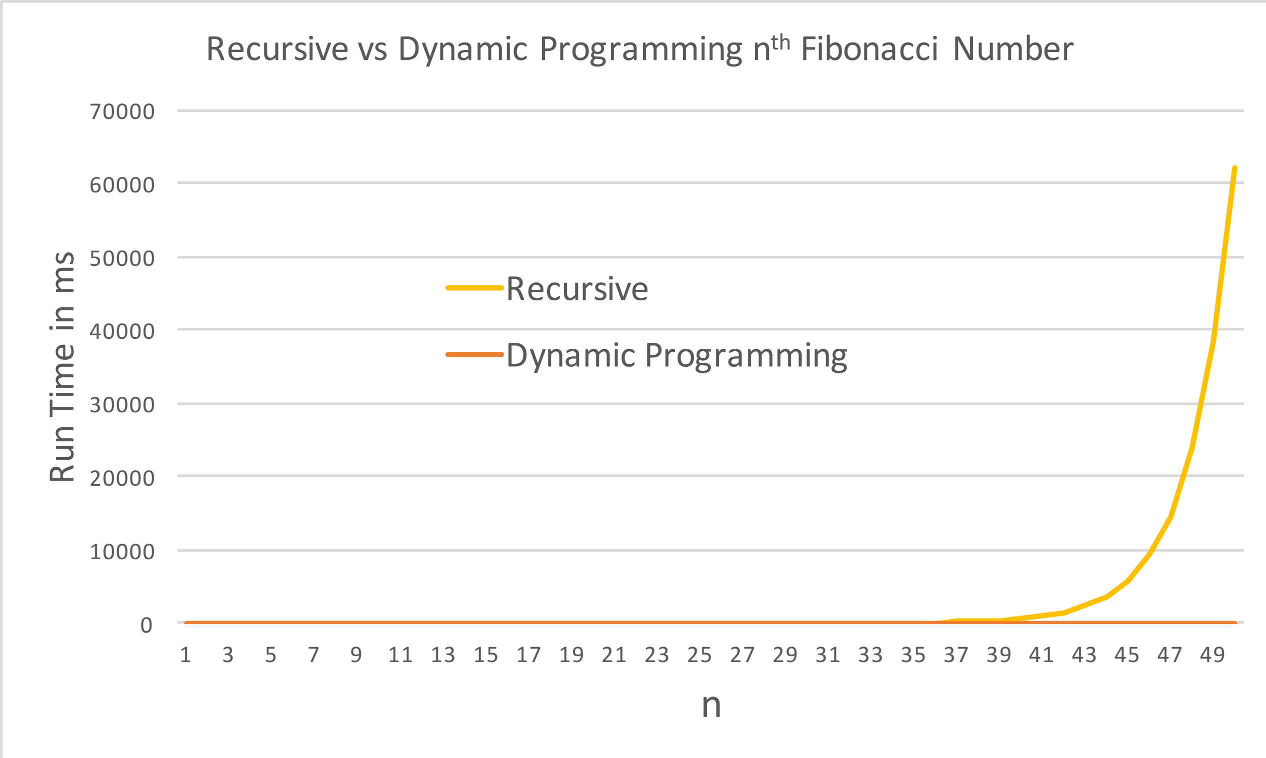
ii. Add F_n to the array by summing the top 2 elements only

iii. 1 for loop of n – 1 elements $\rightarrow O(n)$

4. EXPERIMENTS

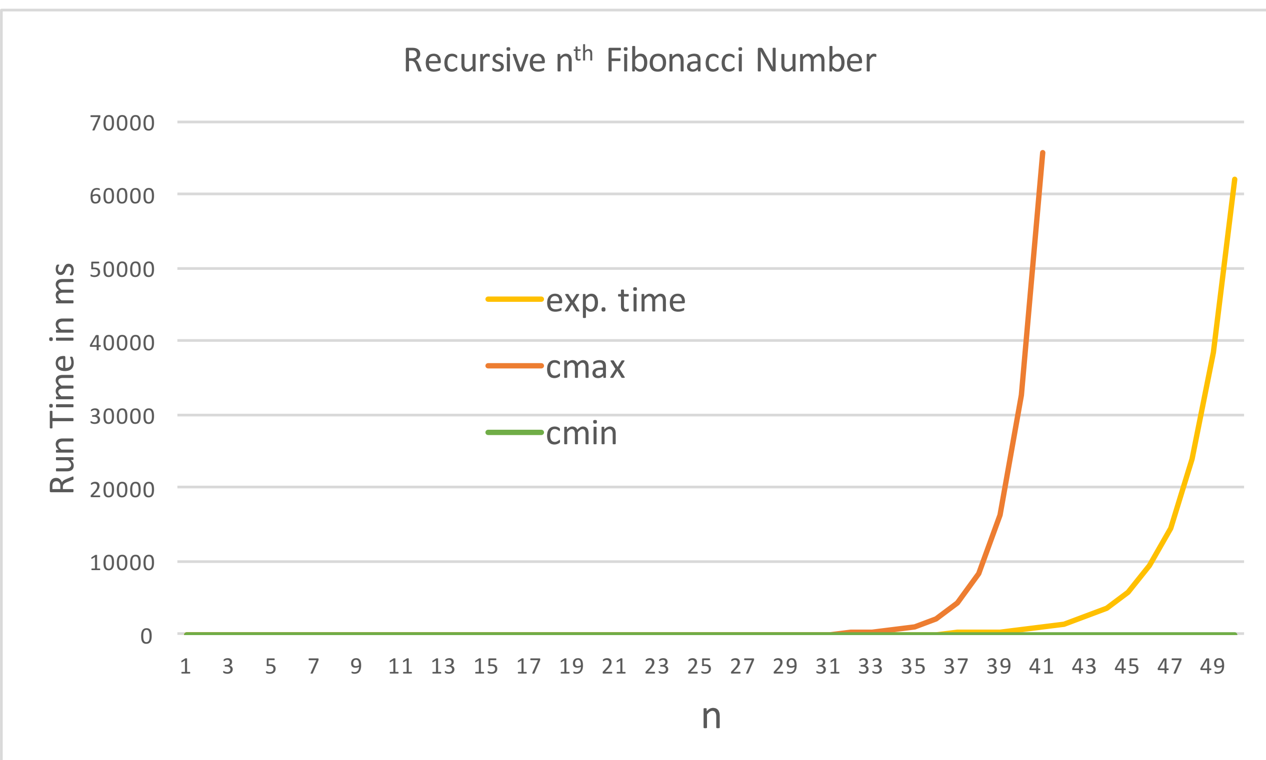
A. Direct comparison

Because the runtimes of the recursive method grow exponentially, the two algorithms can only be directly compared for relatively small values of n. The graph below shows the two algorithms' runtimes in milliseconds from n = 1 up to n = 50. The recursive calculation has a theoretical runtime of $O(2^n)$, while dynamic programming has a theoretical runtime of $O(n)$. Experimental runtimes are shown on the graph below.



B. Recursive calculation

Fibonacci numbers up to F_{50} (n = 50) were calculated and the runtime for each calculation was recorded in milliseconds. For theoretical runtimes, F_0 and F_1 are assumed to take 2^n = 1 and 2ms, respectively. By comparing the experimental runtime to the theoretical runtime, the hidden constant c = (experimental RT / theoretical RT) was calculated. This was then used to bound the experimental runtime, such that $c_{min}2^n$ <= experimental runtime <= $c_{max}2^n$. This function was then graphed for n = 1 to 50.

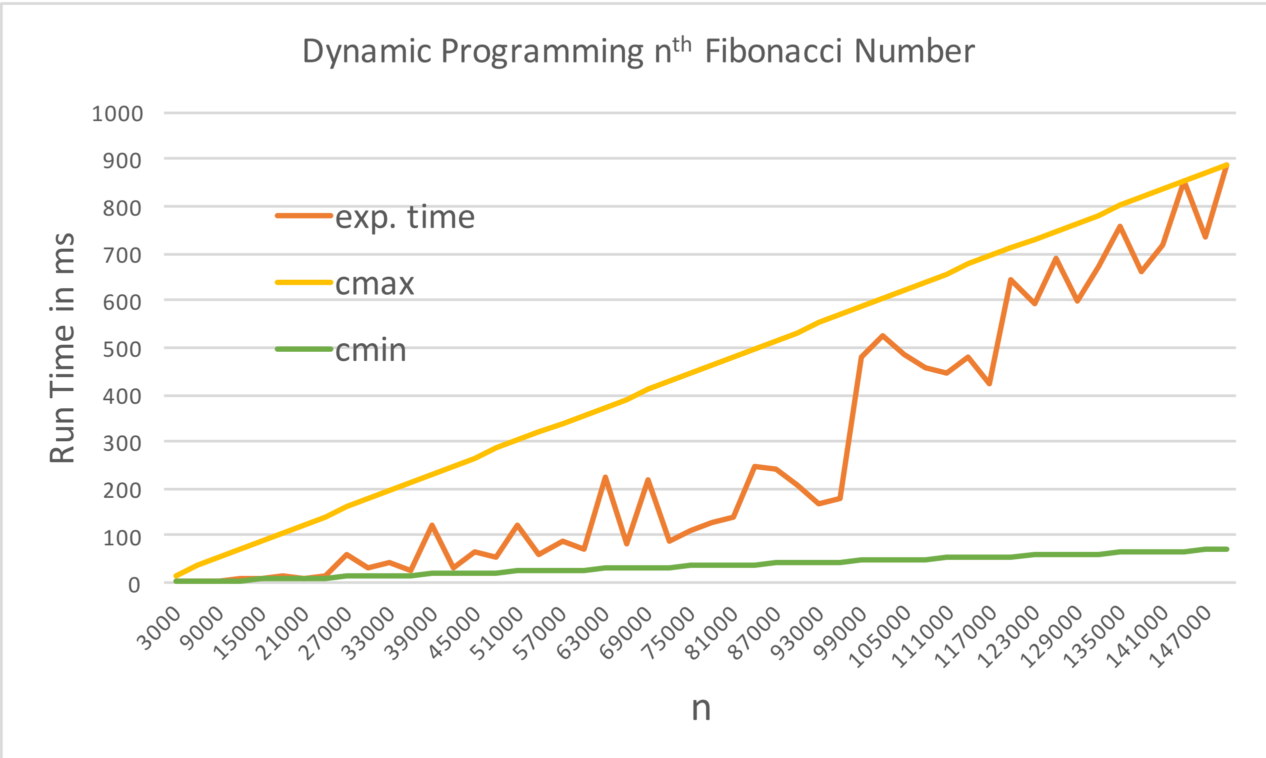


Recursive				
n	value	exp. time	thr. time	c
1	1	0	2	0
2	1	0	4	0
3	2	0	8	0
4	3	0	16	0
5	5	0	32	0
6	8	0	64	0
7	13	0	128	0
8	21	0	256	0
9	34	0	512	0
10	55	0	1024	0
11	89	0	2048	0
12	144	0	4096	0
13	233	0	8192	0
14	377	0	16384	0
15	610	0	32768	0
16	987	0	65536	0
17	1597	0	131072	0
18	2584	0	262144	0
19	4181	0	524288	0
20	6765	0	1048576	0
21	10946	0	2097152	0
22	17711	0	4194304	0
23	28657	0	8388608	0
24	46368	0	16777216	0
25	75025	1	33554432	2.98E-08
26	121393	1	67108864	1.49E-08
27	196418	1	134217728	7.45E-09
28	317811	1	268435456	3.73E-09
29	514229	3	536870912	5.59E-09
30	832040	4	1073741824	3.73E-09
31	1346269	7	2147483648	3.26E-09
32	2178309	11	4294967296	2.56E-09
33	3524578	18	8589934592	2.10E-09
34	5702887	29	1.718E+10	1.69E-09
35	9227465	47	3.436E+10	1.37E-09
36	14930352	77	6.8719E+10	1.12E-09
37	24157817	122	1.3744E+11	8.88E-10
38	39088169	215	2.7488E+11	7.82E-10
39	63245986	321	5.4976E+11	5.84E-10
40	102334155	518	1.0995E+12	4.71E-10
41	165580141	857	2.199E+12	3.90E-10
42	267914296	1382	4.398E+12	3.14E-10
43	433494437	2439	8.7961E+12	2.77E-10
44	701408733	3702	1.7592E+13	2.10E-10
45	1134903170	5615	3.5184E+13	1.60E-10
46	1836311903	9326	7.0369E+13	1.33E-10
47	2971215073	14534	1.4074E+14	1.03E-10
48	4807526976	23867	2.8147E+14	8.48E-11
49	7778742049	38306	5.6295E+14	6.80E-11
50	1.2586E+10	61877	1.1259E+15	5.50E-11

c_{max}	c_{min}
2.98023E-08	0
5.96046E-08	0
1.19209E-07	0
2.38419E-07	0
4.76837E-07	0
9.53674E-07	0
1.90735E-06	0
3.8147E-06	0
7.62939E-06	0
1.52588E-05	0
3.05176E-05	0
6.10352E-05	0
0.00012207	0
0.000244141	0
0.000488281	0
0.000976562	0
0.001953125	0
0.00390625	0
0.0078125	0
0.015625	0
0.03125	0
0.062499999	0
0.124999998	0
0.249999997	0
0.499999993	0
0.999999987	0
1.999999974	0
3.999999948	0
7.999999896	0
15.99999979	0
31.99999958	0
63.99999917	0
127.9999983	0
255.9999967	0
511.9999933	0
1023.999987	0
2047.999973	0
4095.999947	0
8191.999893	0
16383.99979	0
32767.99957	0
65535.99915	0
131071.9983	0
262143.9966	0
524287.9932	0
1048575.986	0
2097151.973	0
4194303.945	0
8388607.891	0
16777215.78	0
33554431.56	0

C. Dynamic Programming calculation

The same methods were applied to the dynamic programming algorithm, but because runtimes grew slowly, change would not be evident unless very large values of n were used. In this case, Fibonacci numbers up to $F_{150,000}$ (n = 150,000) were calculated, starting at n = 3,000 and stepping by 3,000, giving a total of 50 values. For theoretical runtimes, $F_{3,000}$ and $F_{6,000}$ are assumed to take n = 3,000 and 6,000ms, respectively. All other calculation and graphing methods are the same as those used for the recursive calculation, however Fibonacci values are thousands of digits long, so they are truncated in the table.



Dynamic Programming with Big Integers					c_{max}	c_{min}
n	value	exp. time	thr. time	c	0.00592361	0.00047619
3000	4106158863	3	3000	0.001	17.770833	1.42857141
6000	3770131493	5	6000	8.33E-04	35.541666	2.85714282
9000	3461602912	5	9000	5.56E-04	53.312499	4.28571423
12000	3178322757	8	12000	6.67E-04	71.083332	5.71428564
15000	2918224824	9	15000	6.00E-04	88.854165	7.14285705
18000	2679411996	15	18000	8.33E-04	106.624998	8.57142846
21000	2460142407	10	21000	4.76E-04	124.395831	9.99999987
24000	2258816738	17	24000	7.08E-04	142.166664	11.4285713
27000	2073966548	58	27000	0.00214815	159.937497	12.8571427
30000	1904243567	34	30000	0.00113333	177.70833	14.2857141
33000	1748409860	45	33000	0.00136364	195.479163	15.7142855
36000	1605328799	25	36000	6.94E-04	213.249996	17.1428569
39000	1473956772	121	39000	0.00310256	231.020829	18.5714283
42000	1353335570	34	42000	8.10E-04	248.791662	19.9999997
45000	1242585401	66	45000	0.00146667	266.562495	21.4285712
48000	1140898467	55	48000	0.00114583	284.333328	22.8571426
51000	1047533080	124	51000	0.00243137	302.104161	24.285714
54000	9618082471	58	54000	0.00107407	319.874994	25.7142854
57000	8830987023	91	57000	0.00159649	337.645827	27.1428568
60000	8108303504	69	60000	0.00115	355.41666	28.5714282
63000	7444760766	223	63000	0.00353968	373.187493	29.9999996
66000	6835519025	85	66000	0.00128788	390.958326	31.428571
69000	6276134561	221	69000	0.0032029	408.729159	32.8571424
72000	5762527305	89	72000	0.00123611	426.499992	34.2857138
75000	5290951081	109	75000	0.00145333	444.270825	35.7142853
78000	4857966282	128	78000	0.00164103	462.041658	37.1428567
81000	4460414778	138	81000	0.0017037	479.812491	38.5714281
84000	4095396888	249	84000	0.00296429	497.583324	39.9999995
87000	3760250225	240	87000	0.00275862	515.354157	41.4285709
90000	3452530277	208	90000	0.00231111	533.12499	42.8571423
93000	3169992581	170	93000	0.00182796	550.895823	44.2857137
96000	2910576347	181	96000	0.00188542	568.666656	45.7142851
99000	2672389432	478	99000	0.00482828	586.437489	47.1428565
102000	2453694535	523	102000	0.00512745	604.208322	48.5714279
105000	2252896527	483	105000	0.0046	621.979155	49.9999994
108000	2068530817	455	108000	0.00421296	639.749988	51.4285708
111000	1899252669	445	111000	0.00400901	657.520821	52.8571422
114000	1743827392	481	114000	0.0042193	675.291654	54.2857136
117000	1601121337	421	117000	0.00359829	693.062487	55.714285
120000	1470093628	643	120000	0.00535833	710.83332	57.1428564
123000	1349788566	590	123000	0.00479675	728.604153	58.5714278
126000	1239328665	688	126000	0.00546032	746.374986	59.9999992
129000	1137908247	600	129000	0.00465116	764.145819	61.4285706
132000	1044787565	671	132000	0.00508333	781.916652	62.857142
135000	9592874106	759	135000	0.00562222	799.687485	64.2857135
138000	8807841583	661	138000	0.00478986	817.458318	65.7142849
141000	8087052171	719	141000	0.00509929	835.229151	67.1428563
144000	7425248535	853	144000	0.00592361	852.999984	68.5714277
147000	6817603576	734	147000	0.0049932	870.770817	69.9999991
150000	6259685222	887	150000	0.00591333	888.54165	71.4285705

5. PROGRAMMING LANGUAGE

A. Java

i. For handling extremely large numbers, the Long and BigInteger data types must be used
ii. Output to console is too cumbersome, results are saved in a .csv file
iii. Runtimes are determined by taking the current time before and after the calculation, then recording the difference
iv. Smallest runtime granularity is milliseconds

6. SOURCES

Luis, Jose. "Dynamic Programming – Introduction." Java Code Geeks. N.p., 6 Feb. 2014. Web. 07 Mar. 2016. <https://www.javacodegeeks.com/2014/02/dynamic-programming-introduction.html>.

"Program for Fibonacci Numbers - GeeksforGeeks." GeeksforGeeks. N.p., 06 Mar. 2011. Web. 07 Mar. 2016. <http://www.geeksforgeeks.org/program-for-nth-fibonacci-number/>.

"ICS 161: Design and Analysis of Algorithms Lecture Notes for January 9, 1996." Fibonacci Numbers. N.p., 9 Jan. 1996. Web. 07 Mar. 2016. <http://www.ics.uci.edu/~eppstein/161/960109.html>.

"The Fibonacci Quarterly." The Fibonacci Quarterly. Ed. Curtis Cooper. N.p., n.d. Web. 07 Mar. 2016. <http://www.fq.math.ca/index.html>.

Brasch, Thomas Von, Johan Byström, and Lars Petter Lystad. "Optimal Control and the Fibonacci Sequence." J Optim Theory Appl Journal of Optimization Theory and Applications 154.3 (2012): 857-78. Web.

Brock, William A., and Leonard J. Mirman. "Optimal Economic Growth and Uncertainty: The No Discounting Case." International Economic Review 14.3 (1973): 560. Web.

"Bee Ancestry." University Child Development School (2007): n. pag. Web. <http://www.ucds.org/spark/magazine-curriculum/Fibonacci_BeeAncestry.pdf>.

Marshall, Jason. "What Is the Golden Ratio and How Is It Related to the Fibonacci Sequence?" Quick and Dirty Tips. N.p., 5 May 2010. Web. 07 Mar. 2016. <http://www.quickanddirtytips.com/education/math/what-is-the-golden-ratio-and-how-is-it-related-to-the-fibonacci-sequence>.

Cormen, Thomas H., Charles Eric. Leiserson, Ronald L. Rivest, and Clifford Stein. "Fibonacci Heaps." Introduction to Algorithms. Third ed. Cambridge (Mass.): MIT, 2009. 506-30. Print.

Stakhov, Alexey, and Anna Sluchenkova. "Hilbert's Tenth Problem: A History of Mathematical Discovery." Hilbert's Tenth Problem: A History of Mathematical Discovery. Golden Museum, n.d. Web. 07 Mar. 2016. <http://www.goldenmuseum.com/1612Hilbert_engl.html>.