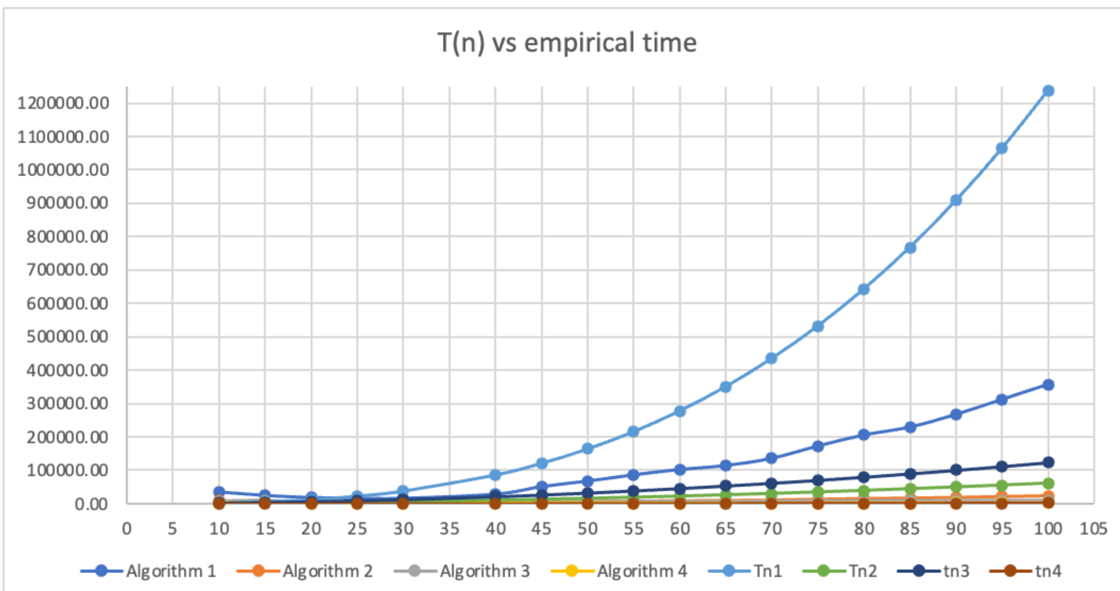
Zach Grindle & Jake Munroe

COMP 3270

Programming Assignment

17 November 2022

Graph of all values



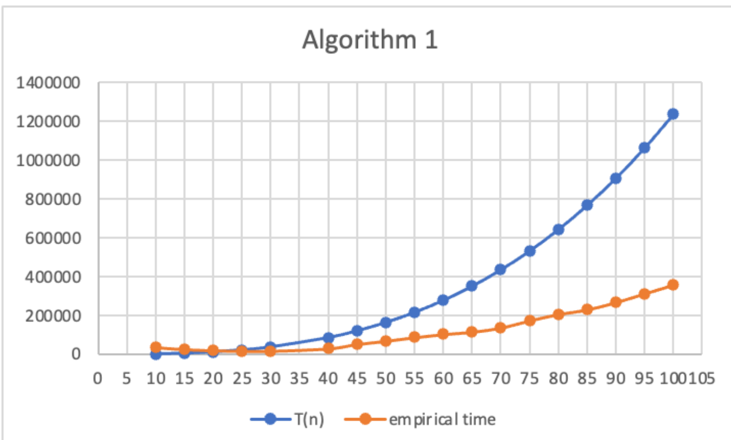
Actual times taken almost always match the predicted time curve. The empirical time typically acts as the upper bound, while the actual runtime is the lower bound. Algorithm 3 acts in the opposite way.

Algorithm-1

Step	Cost of each execution	Total # of times executed
1	1	1
2	1	N+1
3	1	$\sum^{n}_{i=1}$ i+ 1

4	1	$\sum_{i=1}^{n} i$
5	1	$\sum_{i=1}^{n}\sum_{j=1}^{i} J + 1$
6	6	$\sum_{i=1}^{n}\sum_{j=1}^{i} j$
7	5	$\sum_{i=1}^{n} i$
8	2	1

$T_1(n) = (7/6)n^3 + 7n^2 + (41/6)n+6$ $O(n^3)$



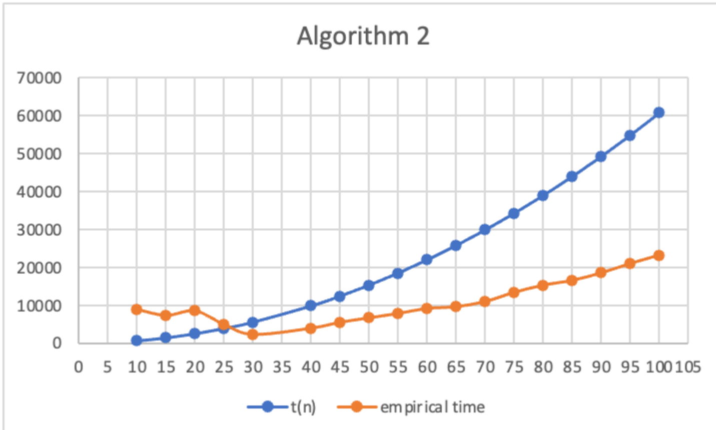
This algorithm's actual time matches the predicted time curve. The predicted is the upper bound for the empirical time. This behavior is as expected.

Algorithm-2

Step	Cost of each execution	Total # of times executed
1	1	1
2	1	N+1
3	1	N
4	1	$\sum_{i=1}^{n} i+1$
5	6	$\sum_{i=1}^{n} i$
6	5	$\sum_{i=1}^{n} i$
7	2	1

$T2(n) = 6n^2 + 7n + 5$

$O(n^2)$



This algorithm behaves as expected, especially after the input 25. The predicted time is the upper bound, while the empirical time is the lower bound.

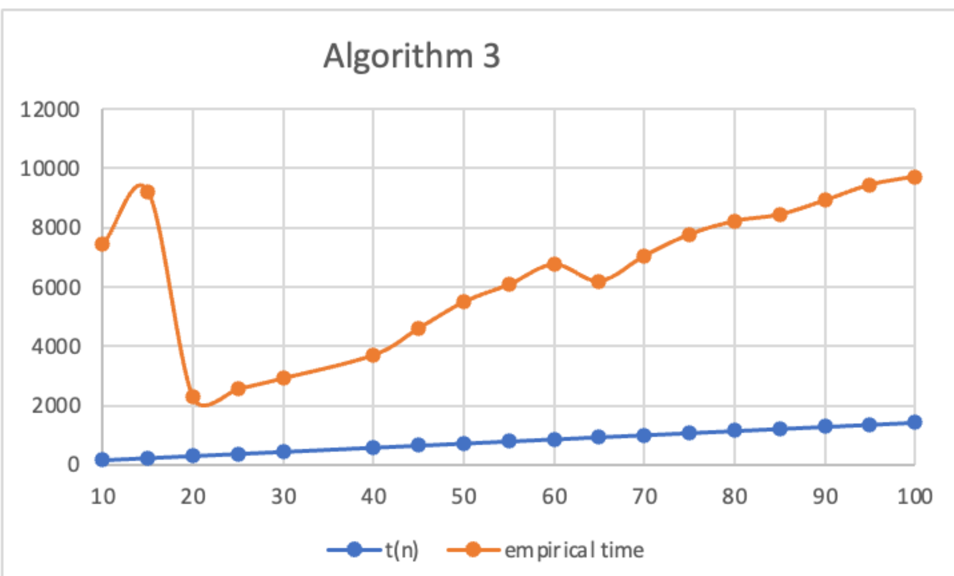
Algorithm-3

Step	Cost of each execution	Total # of times executed in any single recursive call
1	4	1
2	9	1
Steps executed when the input is a base case:		1 or 2
First recurrence relation: T(n=1 or n=0) =		11 for 1 ; 4 for 0
3	5	1
4	2	1
5	1	(n/2) +1
6	6	n/2
7	5	(n/2)
8	2	1
9	1	(n/2) + 1
10	6	n/2
11	5	n/2

12	4	1
13	T(n/2)	(cost excluding the recursive call) log n
14	T(n/2)	(cost excluding the recursive call) log n
15	6	1
Steps executed when input is NOT a base case:1 through 15		
Second recurrence relation: T(n>1) =2T(n/2) + c		
Simplified second recurrence relation (ignore the constant term): T(n>1) =(2^i(T))(n/2^i) + ((2^i) -1)		

Solve the two recurrence relations using any method (recommended method is the Recursion Tree). Show your work below:

$T_3(n)$ =14n+ logn +23 O(n log n)



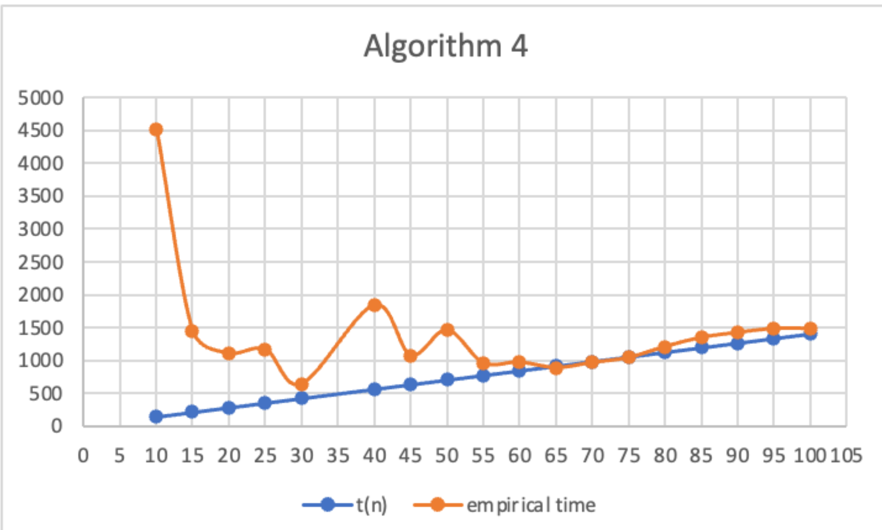
This graph behaved opposite of the other 3, with T(n) being the lower bound, and the empirical time being the upper bound.

Algorithm 4

Step	Cost of each execution	Total # of times executed
1	1	1

2	1	1
3	1	N + 1
4	8	n
5	5	n
6	2	1

T4(n) =14n + 5 O(n)



Algorithm 4 displayed unexpected behavior. Empirical time does not follow the predicted time curve until around the value 55. After this, the empirical time gradually matches the predicted time.