1. Find C1 and C2 and n0 to show that 5x2 + 4x -8320 is Θ(n2)

Solution:

1. An algorithm can process 512 items in 2 seconds

6144/514 = 12 🡪 12t

* 1. Time for 6144 items if algorithm is is Θ(n2)

2\*122 = 288 s

* 1. Time for 6144 items if algorithm is is Θ(n)

2\*12 = 24 s

* 1. Time for 6144 items if algorithm is is Θ(n3)

2\*123 = 3456 s

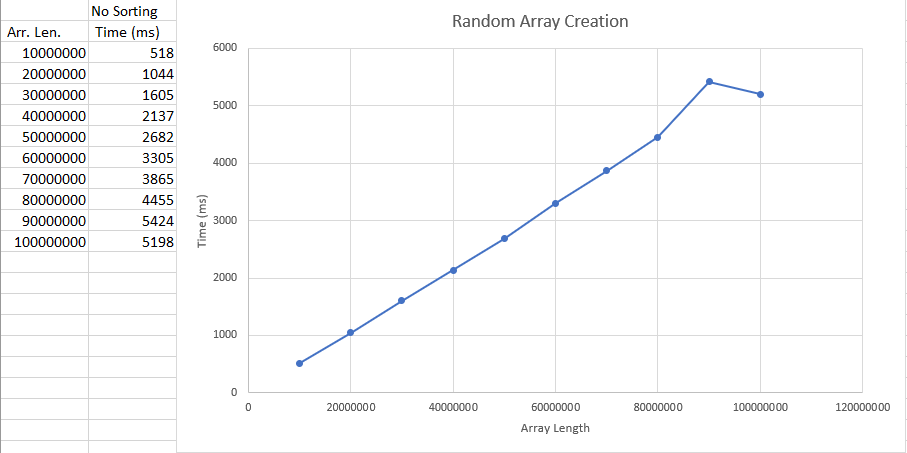
* 1. Time for 6144 items if algorithm is is Θ(2n)

2\*212 = 213 s

* 1. Time for 6144 items if algorithm is is Θ(n1/2)

2\*sqrt(12) = 6.928 s

1. Write a program – See Attached



This is an O(n) algorithm whose completion time y (milliseconds) can be predicted by:

y = 0.00005626485\*x - 71.26667, where x is the length of the array

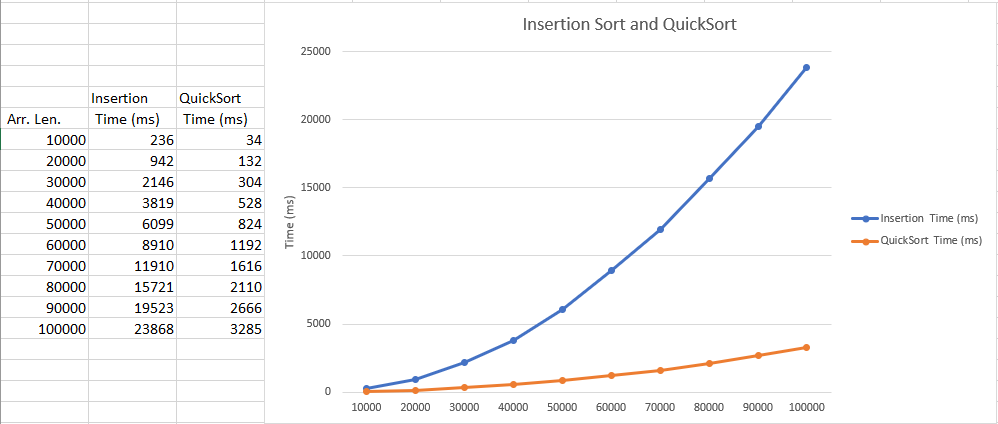
To solve for items generated in 3 days, we must solve for X where Y = 3days

1000ms/s\*60s/m\*60m/hr\*24hr/d\*3d = 259200000ms

259200000 = 0.00005626485\*x - 71.26667

**X = 4,606,785,199,759 numbers**

1. Insertion Sort



This is an O(n2) algorithm whose completion time y (milliseconds) can be predicted by:

y = -234.4 + 0.01334788\*x + 0.000002290303\*x2, where x is the length of the array

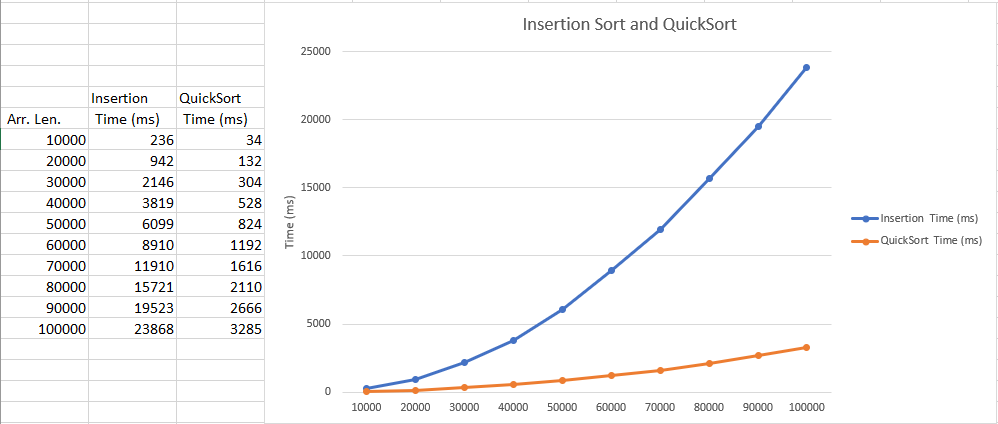
To solve for items generated in 3 days, we must solve for X where Y = 3days

3 days = 1000ms/s\*60s/m\*60m/hr\*24hr/d\*3d = 259200000ms

259200000 = -234.4 + 0.01334788\*x + 0.000002290303\*x2

**X = 10,635,362 numbers**

1. QuickSort



This is a O(n\*log(n)) algorithm. A 3 day approximation is harder to predict via a regression extrapolation.

We need to model a function f(n) = C\*n\*log(n)

Solving for c for y=23868, x=100000… C =

23868 = C \* 100000\* log2(100000)

C = 0.01436996787

We can get a rough estimate for the array length by solving for x:

259200000ms = 0.01436996787\*x\*log(x)

**X = 876,000,000 numbers**

1. Given the table, fill in the rest of the bounds

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 🡨 | Problem | 🡪 | 🡨 | Algorithm | 🡪 | 🡨 | Solution | 🡪 |
| Best | Average | Worst | Best | Average | Worst | Best | Average | Worst |
|  |  |  |  | Ω(n) | O(n) |  |  |  |
|  |  |  | Θ(n2) |  |  |  | O(n2) |  |
| Ω(n2) |  |  |  | Θ(n2) |  |  | Ω(n2) |  |