NOTE: please note that I used seconds not milliseconds in my timing

REPORT

1. The asymptotic upper bound for running time of quick sort is

O(n^2). The recurrence is below.

T(n) = n+ T(n-1) ; n>1 -🡪 equation 1

1. ; n=1

T(n-1) = (n-1)+T(n-2) ---🡪 equation 2

T(n-2) = (n-2)+T(n-3) ---🡪 equation 3

Substitute equation 2 in 1

T(n) = n + T(n-1)

T(n) = n+(n-1) + T(n-2) ---equation 4

Now substitute equation 3 in 4

T(n) = n+ (n-1)+(n-2)+(n-3)

So the pattern is

T(n) = n…….+(n-k)+T(n-(k+1))

We stop we when:

n-(k+1)=1

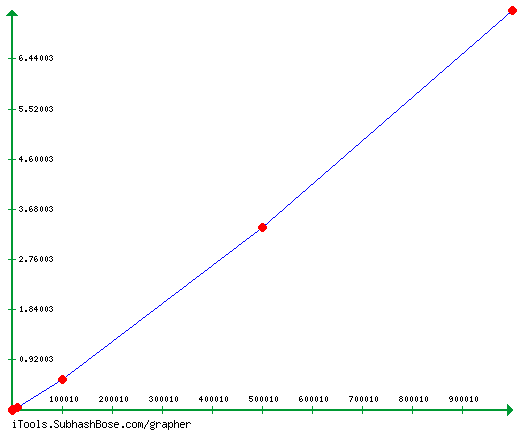
k=n-2

n….+(n-n+2)+T(n-(n-2+1))

= n…..+2+1 = (n(n+1))/2 = 0(n^2)



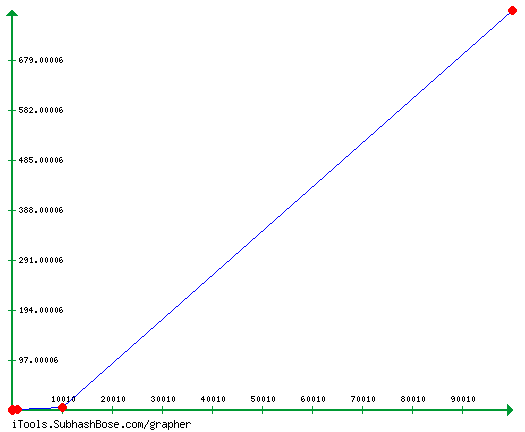
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No of residence | 10 | 100 | 1000 | 10000 | 100000 | 500000 | 100000 |
| Time(sec) | 0.00003 | 0.000209 | 0.0003735 | 0.052214 | 0.5536 | 3.3491 | 7.339 |



Comment: As we can see the time unit increases as we increase the input size but the sort that we did here is base on an average case and in contrast with the case of what I found in (a) which is n^2 , this one that I plotted in base on average case which is measured(omega n(logn)).

c. The worst case for quick sort is O(n^2). This worst case happens when we are unlucky and our pivot element is always chosen as the smallest or the biggest

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No of residence | 10 | 100 | 1000 | 10000 | 100000 | 500000 | 100000 |
| Time(sec) | 0.00006 | 0.00033 | 0.0612 | 4.245 | 776.2 | ? | ? |



The first solution that I propose is that we can randomly select a pivot (randomize select) while my second solution is that we can choose the median of three, meaning we select three elements (usually first, middle and last), then we chose there median as the pivot.

d. Quick sort is not a stable algorithm, the explanation is that let say we were asked to sort only 4 residence which are as follows.

before sorting

1 80 84 females 80612 8600000US80612

0 15 17 female 27982 8600000US27982 🡪 p1

5 85 53 males 64120 8600000US64120

0 80 84 females 67639 8600000US27982 ->p2

after sorting base on population and geo\_id, we can have

0 15 17 female 27982 8600000US27982 🡪 p1

0 80 84 female 67639 8600000US27982 ->p2

1 80 84 female 80612 8600000US80612

5 85 53 male 64120 8600000US64120

but when we sort again ,we can end up with

0 80 84 female 67639 8600000US27982 ->p2

0 15 17 female 27982 8600000US27982 🡪 p1

1 80 84 female 80612 8600000US80612

5 85 53 male 64120 8600000US64120

so the order of p1 and p2 is inconsistent