Analysis of ~~Merge~~ Quick sort.

$T(0) = T(1) = 1$ based on no of comparison for n=0 or 1

Running time of Quick sort with n elements = Running time of two recursive calls + linear time spent in partition.

(with the assumption that pivot selection takes constant time)

If $|S| = n$ and $|S_1| = i$

$\Rightarrow |S_2| = (n-i-1)$

So $T(n) = T(|S_1|) + T(|S_2|) + cn$

$$\Rightarrow T(n) = \begin{cases} 1 & ; n=1 \; T(1)=1 \\ T(i) + T(n-i-1) + cn & ; n>1 \end{cases}$$

$T(n) = T(i) + T(n-i+1) + cn$

Worst case Analysis

If the pivot selected is smallest element (largest element) all the time then $i=0$

that is $|S| = n$ Input set

After first pass $\{|S_1| = 0\}$ & $\{|S_2| = n-1\}$

So the recurrence relation takes the form

$T(n) = T(n-1) + cn \quad n>1$

as $T(0) = 1$ is ignored for $i=0$

The equation will be solve by using method of telescopic sum

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

$T(n-1) = T(n-2) + c(n-1)$

$T(n-2) = T(n-3) + c(n-2)$

$\vdots$

$T(2) = T(1) + c(2)$

Adding up all these equations yields

$$T(n) = T(1) + c\sum_{i=2}^{n} i$$

$\Rightarrow T(n) = O(n^2)$ as $\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \approx n^2/2$

QUICKSORT Average case Analysis

Let us assume that in a given set of n element, selecting any element as pivort is equally likely

i.e. $1/n$

Hence the file size for S_1 is equalily likely, and

$|S_1| = 0$ or 1 or 2 or $\cdots$ $(n-1)$

Let $|S_1| = i$

then

Time complexity of Quick SORT

$T(n) = T(i) + T(n-i-1) + cn$

So average value of $T(i)$ can be given as

$$T(i) = \frac{1}{n} \sum_{j=0}^{n-1} T(j)$$

$T(0)$ $T(n-1)$
$T(1)$ $T(n-2)$
$T(2)$ $T(n-3)$
$\vdots$ $\vdots$
$T(n-1)$ $T(0)$

All possible sub problem →

⇒ Total sub problems

$$= 2 \sum_{i=0}^{n-1} T(i)$$

$$T(n) = \frac{2}{n} \left[\sum_{j=0}^{n-1} T(j) \right] + cn$$

$$T(n) = \frac{2}{n} \left[\sum_{j=0}^{n-1} T(j) \right] + cn$$

Multiplying the above equation by n

$$* \quad nT(n) = 2 \sum_{j=0}^{n-1} T(j) + cn^2$$

In order to remove summation sign from the above equation we telescope with one more equation

$$** \quad (n-1)T(n-1) = 2 \sum_{j=0}^{n-2} T(j) + c(n-1)^2$$

Subtracting ** from *

$$nT(n) - [(n-1)T(n-1)]$$

$$= \left[2 \sum_{j=0}^{n-1} T(j) + cn^2 \right] - \left[2 \sum_{j=0}^{n-2} T(j) + c(n-1)^2 \right]$$

$$\Rightarrow nT(n) - (n-1)T(n-1) = 2T(n-1) + cn^2 - cn^2 - c + 2cn$$

$$\Rightarrow nT(n) - (n-1)T(n-1) = 2T(n-1) + 2cn - c$$

by dropping the insignificant term $-c$

$$\Rightarrow nT(n) = 2T(n-1) + (n-1)T(n-1) + 2cn$$

Quick Sort — Average case Analysis

$\Rightarrow n\,T(n) = (n+1)\,T(n-1) + 2cn$

Now conditioning of above equation so that it can be solved using the method of telescopic sum to find the soln of recurrence relation.

$n\,T(n) = (n+1)\,T(n-1) + 2cn$

dividing the above eqn by $n(n+1)$

$$\Rightarrow \frac{n\,T(n)}{n(n+1)} = \frac{(n+1)\,T(n-1)}{n(n+1)} + \frac{2cn}{n(n+1)}$$

$$\Rightarrow \frac{T(n)}{(n+1)} = \frac{T(n-1)}{n} + \frac{2c}{(n+1)}$$

Now apply telescopic Sum method

$$\frac{T(n-1)}{n} = \frac{T(n-2)}{(n-1)} + \frac{2c}{n}$$

$$\frac{T(n-2)}{(n-1)} = \frac{T(n-3)}{(n-2)} + \frac{2c}{n-1}$$

$$\vdots$$

$$\frac{T(2)}{3} = \frac{T(1)}{2} + \frac{2c}{3}$$

Adding the above Eqns

$$\frac{T(n)}{(n+1)} = \frac{T(1)}{2} + 2c\sum_{i=3}^{n+1} \frac{1}{i}$$

Sum?

Assumption

The sum $2c\sum_{i=3}^{n+1} \frac{1}{i}$ is about $\log_e(n+1) + \nu - 3/2$

where $\nu \approx 0.577$ & known as Euler's const.

So $\frac{T(n)}{(n+1)} = O(\log n)$

$\Rightarrow T(n) = (n+1)\,O(\log n)$

$\Rightarrow T(n) = O(n \log n)$

Q. present an analysis of quick sort when all keys are EQUAL.

Refer the Deterministic Algorithm of Quick sort.

Select the 1st item as pivort.

after 1st pass we have

$|S_1| = 0$ & $|S_2| = n-1$ ✓

~~OR~~

S_1

The complexity performance depends on PARTITION algorithm

Best case Analysis

If the pivot selected in each pass divides the array exactly equal halfs, that leads to the best case

[Assumption: We assume that the two subarrays are each exactly half the size of the original, and although this gives a slight over estimate,

This is acceptable because we are looking for the answer in a Big-oh notation (upper bound only)

The Time complexity

$T(n) = 2T(n/2) + cn$

By dividing n

$T(n)/n = T(n/2)/(n/2) + c$ ... (1)

Using telescopic sum method to above equation

Let $n = 2^k$

$\Rightarrow \log n = \log 2^k$

$\Rightarrow \log n = k \log 2$

$\Rightarrow k = \log n$

$T(n)/n = T(n/2)/(n/2) + c$

$T(n/2)/(n/2) = T(n/4)/(n/4) + c$

$T(n/4)/(n/4) = T(n/8)/(n/8) + c$

$\vdots$

$T(2)/2 = T(1)/1 + c$

k equations as $n = 2^k$

Adding above equations (kc = clogn)

$T(n)/n = T(1) + c\log n$

$\Rightarrow T(n) = n + cn\log n$ as $T(1) = 1$

Hence $T(n) = O(n \log n)$

Analysis of Quickselect:

How Quick sort is used to find the solution to selection problem [Find the K^{th} LARGEST (Smallest)]

If the pivot selected is placed at k^{th} position in the array[A0] then that results the best case of the selection problem

if not it is possible to discarde one of the S1 or S2 [Two partitions]

that includes k^{th} element of the array; and apply Quick sort to S1 or S2; Results recurrance relation as follows.

$$T(n) = T(i) \text{ or } T(n-i-1) + cn$$

$$T(n) = \frac{1}{n}\sum_{j=0}^{n-1} T(j) + cn$$

$$T(n) = \frac{1}{n}\sum_{j=0}^{n-1} T(j) + cn$$

$$T(n-1) = \frac{1}{(n-1)}\sum_{j=0}^{n-2} T(j) + c(n-1)$$

$$nT(n) = \sum_{j=0}^{n-1} T(j) + cn^2 \quad \text{——} (1)$$

$$(n-1)T(n-1) = \sum_{j=0}^{n-2} T(j) + c(n-1)^2 \quad \text{——} (2)$$

By ~~adding~~ subtracting Eq 1 and Eq 2

$$nT(n) - [(n-1)T(n-1)] = T(n-1) + cn^2 - cn^2 + c - 2cn$$

$$\Rightarrow nT(n) = [T(n-1)(n-1)] + T(n-1) - c + 2cn$$

By dropping c

$$\Rightarrow nT(n) = T(n-1)[n-1+1] + 2cn$$

$$\Rightarrow nT(n) = nT(n-1) + 2cn$$

$$T(n) = T(n-1) + 2c$$

Apply Telescupic sum

$$T(n) = T(n-1) + 2c$$
$$T(n-1) = T(n-2) + 2c$$
$$T(n-2) = T(n-3) + 2c$$
$$\vdots$$
$$T(2) = T(1) + 2c$$

Adding all equations

$$T(n) = T(1) + 2cn$$

Small problem $T(1) = 1$

$$\Rightarrow T(n) = 1 + 2cn$$

$$\Rightarrow T(n) = O(n)$$

Time complexity Average case for

Algorithm Quickselet → in next page