15 September 2023

runtime of Quicksort (cont.)

$$T(n) = \max_{1 \le i \le n} \{T(i-1) + T(n-i)\} + \Theta(n)$$

on extremes: $i = 1$ or n

$$T(n) = T(0) + T(n-1) + \Theta(n) \in \boxed{\Theta(n^2)}$$ ← worst-case behavior

in middle:

$$T(n) \approx T(\tfrac{n}{2}) + T(n/2) + \Theta(n)$$
$$= 2T(n/2) + \Theta(n) \in \Theta(n \log n)$$ ← wouldn't this be nice

But,... let's consider the average case analysis

Note: $T(n)$ is the same (asymptotically) as counting the number of comparisons.

Random variable: $X_{ij} = \begin{cases} 1, & \text{if els } i + j \text{ are compared} \\ 0, & \text{otherwise} \end{cases}$

The total # of comparisons (& hence the RT) is:

$$X = \sum_{i=1}^{n} \sum_{j=i+1}^{n} X_{ij}$$

What is the expected value of this?

$$\mathbb{E}(X) = \mathbb{E}\Big(\sum_{i=1}^{n} \sum_{j=i+1}^{n} X_{ij}\Big)$$
$$= \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathbb{E}(X_{ij}) \quad \text{by linearity of expectation}$$
$$= \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \quad \text{(see "timeline")}$$
$$= \sum_{i=1}^{n} 2 \cdot \Big(\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-i+1}\Big) \le \sum_{i=1}^{n} 2 H_n = 2n H_n$$
$$\le 2n \ln(n) = \Theta(n \log n).$$

2

Recall (from long ago) the Harmonic Numbers:

$$\sum_{i=1}^{n} 1/i = H_n \in [\ln(n), \ln(n)+1]$$
$$\in \Theta(\ln(n)) = \Theta(\log n)$$

Geometric Series

$$\sum_{i=1}^{\infty} a \cdot r^i$$

→ if $r \geq 1$, diverges

→ if $0<r<1$, it converges to $a\left(\frac{1}{1-r}\right)$.

Why is QS correct? Use our recursion invariants?

(1) State our assumptions needed for a valid call to QS.

a) A is an array of real #s
b) n is the length of A

(2) State the Recursion Invariant:
QS(A) executes correctly if:
a) A is sorted from smallest to largest

(3) INITIALIZATION
- identify our base case: $n=0$
 (can't make a recursive call in our base case)

Proof:
Since $n=0$, A is empty.
In the code, I just return. Now I'm done.
Since A is still empty, (2a) is vacuously true. ∎

Note: this base case might not be the 1st call.
There could be another "A" that our empty array is a subarray of.

Hur-ray! INIT cases are almost always this "easy".

④ MAINTENANCE: Assuming the recursion fairy is always correct, we prove that a general call is correct.

Let $\boxed{n_1 \geq 0}$.

Assume the recursion fairy works for QS(A) when $|A| \leq n_1$. That is (restate RI), after QS(A) executes, A is sorted from smallest to largest.

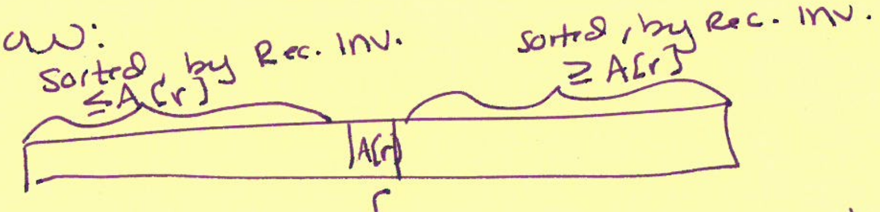
Now, we're given a call QS(A), where $|A| = n_1 + 1$ (ie, slightly bigger than my recursion fairy can handle, so I've gotta do this one)

Since this is a valid call, I know by ① that A is an array of real #s.

Since $n_1 \geq 0 \Rightarrow n_1 + 1 \geq 1$. Thus, we evaluate TRUE on line 1. p is our pivot index chosen on line 3. Since $\boxed{\text{we proved}}$ PARTITION (A, p) will return r, the new index of A[p] from the input A in the new order of A, and that A[1 ... r-1] are all ≤ A[r] and A[r+1 ... n] are all ≥ A[r] and A has the same #s, just in a diff order than initially given.

Well, not yet, but HZ...

Then, the RF solves lines 4 + 5 for me, so I know:



∴ the whole array A is sorted!
(which is exactly what I wanted to prove) from 2a