EECS3101 notes

Jerry Wu

2023-02-07

Sorting continued:: Quicksort

> *"Tony Hoare publically apologized for inventing NullPointerException"- Larry YL Zhang 2023*

The main idea::

- Pick a pivot and partition the array (usually the last one).
- Left side $\equiv$ less than or equal to pivot, right side $\equiv$ greater than or equal to pivot.
- Recursively partition until the sub array only has one element, then the array is sorted.

Worst case analysis of quicksort

$T(n)$ is the total number of comparisons made. We first prove that $T(n) = O(\xi)$, where ξ is something. Then we prove $T(n) = \Omega(\xi)$, and finally $T(n) = \Theta(\xi)$

Consider the array::

$$A = \{2, 1, 3, 4, 7, 5, 6, 8\}$$

Claim 1:: Suppose A is some array, we claim that every $x \in A$ can only be chosen once. ($O(n)$)

Claim 2:: Elements are only compared to pivots (this is what partitioning into sub arrays does).

Claim 3:: Any pair $(a, b) \in A$ are only compared with each other **at most once**. The only possible one happens when a or b is chosen as a pivot and the other is compared to it. But after being the pivot, the pivot will be out of the market and never compared with anyone anymore.

> *"One element goes into the recursive call, and the other is the pivot. They will never see each other again in their lifetimes. It's a bit sad when you think of it like that."-Larry YL Zhang 2023*

So the total number of comparisons is **no more than the total number of pairs!** We can express it as the following::

$$T(n) \leq_n C_2 = \frac{n(n-1)}{2} \implies T(n) \in O(n^2)$$

We want to show that

$$\exists x, t(x) \geq Cn^2$$

Now, we show that $T(n) \in \Omega(n^2)$, i.e. the worst case is lower bounded by some Cn^2. We can find a number that results in a bad partition (aka the biggest number in the array). Ironically, the **worst case** for quick sort is an already sorted array, since the pivot (last index) will be the max value. We can now say that::

$$T(n) \geq \frac{n(n-1)}{2} \implies T(n) \in \Omega(n^2) \implies T(n) \in \Theta(n^2)$$

It is $\Theta(n^2)$ since we proved both $O(n^2)$ and $\Omega(n^2)$.

Average case analysis of quicksort (VERY COMPLICATED)

Input:: Any permutation of array and each permutation has equal probability (uniform randomness).
After some trivial math stuff, we have that the average case performance is $O(nlog(n))$! (chapter 7.4 CLRS textbook).

So is quick sort really quick? Yes, in average case.

> *"Hackers will always send a sorted input to make your code slow"-Larry YL Zhang 2023*

However...

In reality, it's impossible to assume uniform randomness since it is impossible for us to know what the input distribution really is. Hackers can also sabotage input and always put sorted inputs into the algorithm. We can solve this problem by randomization. This will ensure that the input is as close to uniform randomization as possible.

> *"Randomization is like a cup of coffee. We can't guarantee the exact flavour of the coffee, but the average taste is always the same."-Larry YL Zhang 2023*

> *"For some reason, all my examples are related to food and drink"-Larry YL Zhang 2023*

Example time

Consider the partition method for quicksort. $Partition(A, start, end)$. Develop a loop invariant that performs the method in place (slide 25).

Example 2

Write all possible arrays that store the binary max heap with the keys:: $1, 2, 3, 4, 5$

> *"Don't do it with BOGO sort"-Larry YL Zhang 2023*
>
> *"That number seems a bit small to me"-Larry YL Zhang 2023*

We kow that 5 always has to be the root, from there, we arrange each element under 5. We get 6 possibilities in total for the first case, and 2 possibilities for the 2nd case. So we have 8 in total.