CSE331: Introduction to Algorithms
Notes on Lecture 12: Quicksort

Antoine Vigneron

July 23, 2021

1 Assumption that input numbers are distinct

If the input numbers are not distinct, randomized QUICKSORT may run in quadratic time. Suppose for instance that all input numbers are the same, that is $A[i] = A[n]$ for all $i = 1, \ldots, n$. Then the "yellow" array constructed by PARTITION is empty as it contains elements stricly larger than $A[n]$. Then the running time is given by $T(n) = T(n-1) + \Theta(n)$, which yields $T(n) = \Theta(n^2)$.

2 Quadratic upper bound

In this section, we prove the upper bound $T(n) = O(n^2)$ using the substitution method. We showed that $T(n)$ satisfies the recurrence relation below:

$$T(n) \leqslant \max_{0 \leqslant q \leqslant n-1} (T(q) + T(n-q-1)) + \Theta(n).$$

It means that there is a function $f(n) = \Theta(n)$ such that

$$T(n) \leqslant \max_{0 \leqslant q \leqslant n-1} (T(q) + T(n-q-1)) + f(n).$$

As $f(n) = \Theta(n)$, there exists a constants $d > 0$ such that $f(n) \leqslant dn$ for all $n \geqslant 1$. It follows that

$$T(n) \leqslant \max_{0 \leqslant q \leqslant n-1} (T(q) + T(n-q-1)) + dn \tag{1}$$

for all $n > 1$.

As we observed that Quicksort runs in $\Theta(n^2)$ time when the input array is sorted, we guess that the solution is $T(n) = O(n^2)$. So we want to prove that $T(n) \leqslant cn^2$ for some constant c. We make a proof by induction. In order to handle the **base case** $T(1) \leqslant c.1^2$, we need to choose $c \geqslant T(1)$.

We now prove the **inductive step**. So we assume that $n > 1$, and $T(m) \leqslant cm^2$ for all $m < n$, and we want to show that it implies $T(n) \leqslant cn^2$. It follows from our inductive hypothesis that

$$\begin{aligned} T(n) &\leqslant \max_{0 \leqslant q \leqslant n-1} (T(q) + T(n-q-1)) + dn \\ &\leqslant \max_{0 \leqslant q \leqslant n-1} \left(cq^2 + c(n-q-1)^2\right) + dn. \end{aligned}$$

The expression $cq^2 + c(n-q-1)^2$ is a degree-2 polynomial in q, and the leading coefficient is positive. So it achieves its maximum at the boundary of the interval $[0, n-1]$, hence its maximum is $c(n-1)^2$, and we have

$$\begin{aligned} T(n) &\leqslant c(n-1)^2 + dn \\ &= cn^2 + (d-2c)n + c. \end{aligned}$$

If we choose $c \geqslant d$, then we have $d - 2c \leqslant -c$. As we assumed that $n > 1$, it implies that $(d-2c)n + c \leqslant 0$ and thus

$$T(n) \leqslant cn^2.$$

It proves the inductive step, under the assumption that $c \geqslant d$. We also need $c \geqslant T(1)$ in order to satisfy the base case. So we complete the proof by choosing $c = \max(d, T(1))$.