Assignment 4

Benjamin Jakubowski

June 22, 2016

1 Illustrating randomized quicksort

We illustrate the operation of randomized quicksort on the array

$$A = [19, 2, 11, 14, 7, 17, 4, 3, 5, 15]$$

by showing the values in array A after each call to `random_partition`. Note the index is 0-indexed (since randomized quicksort was implemented in Python).

Time	**Time**	**State**
Initial State	NA	A = [19, 2, 11, 14, 7, 17, 4, 3, 5, 15]
After call 1	Pivot = 15 at position = 7	A = [2, 11, 14, 7, 4, 3, 5, 15, 19, 17]
After call 2	Pivot = 5 at position = 3	A = [2, 4, 3, 5, 11, 14, 7, 15, 19, 17]
After call 3	Pivot = 3 at position = 1	A = [2, 3, 4, 5, 11, 14, 7, 15, 19, 17]
After call 4	Pivot = 7 at position = 4	A = [2, 3, 4, 5, 7, 14, 11, 15, 19, 17]
After call 5	Pivot = 11 at position = 5	A = [2, 3, 4, 5, 7, 11, 14, 15, 19, 17]
After call 6	Pivot = 17 at position = 8	A = [2, 3, 4, 5, 7, 11, 14, 15, 17, 19]
Final array	NA	A = [2, 3, 4, 5, 7, 11, 14, 15, 17, 19]

2 Longest and shortest path in quicksort recursion tree

Let $0 < \alpha \leq 1/2$ be a constant split proportion in the quicksort recursion tree (such that at every level of quicksort the splits are in the proportion $1-\alpha$ to α). Then the shortest path in the recursion tree is the path getting α of the work on each split, and the longest path in the tree is the path getting $1-\alpha$ of the work on each split. Now note the tree grows until $p = r$, i.e. the leaf nodes have a single (trivially sorted) subarray.
Thus, the longest path is length ℓ where

$$
\begin{aligned}
& n(1-\alpha)^{\ell} \approx 1 \\
\Longrightarrow \quad & (1-\alpha)^{\ell} \approx 1/n \\
\Longrightarrow \quad & \ell \log(1-\alpha) \approx \log(1/n) \\
\Longrightarrow \quad & \ell \approx -\log(n)/\log(1-\alpha)
\end{aligned}
$$

So (ignoring integer roundoff), we have

$$\ell \approx -\log(n)/\log(1-\alpha)$$

Similarly, the shortest path is of length ℓ where

$$
\begin{aligned}
& n(\alpha)^{\ell} \approx 1 \\
\Longrightarrow \quad & (\alpha)^{\ell} \approx 1/n \\
\Longrightarrow \quad & \ell \log(\alpha) \approx \log(1/n) \\
\Longrightarrow \quad & \ell \approx -\log(n)/\log(\alpha)
\end{aligned}
$$

So (again ignoring integer roundoff) we have

$$\ell \approx -\log(n)/\log(\alpha)$$

3 Probability of a split more balanced than $1-\alpha : \alpha$

Let $A = [a_1 \cdots a_n]$ be a random array. Then let $\pi = [\pi_1 \cdots \pi_n]$ be the permutation that stably sorts A (i.e. such that $Z = [A[\pi_1] \cdots A[\pi_n]]$ is sorted). Then (letting r be the pivot index), since A is random, the probability $A[r] = A[\pi_i] = 1/n$ for all $i \in \{1, \cdots, n\}$.

Now let $A[r] = A[\pi_j]$ be fixed (for some $j \in \{1, \cdots, n\}$). Then the splits produced by `partition` are *as or less* balanced than those produced by pivoting on $A[r] = A[\pi_k]$ for

$$k \in \{1, \cdots, \min\{j, n-(j-1)\}\} \cup \{\max\{j, n-(j-1)\}, \cdots, n\} = S$$

Note we let S represent this set to ease notation.

Thus, the probability of `partition` producing splits that are *as or less* balanced than those obtained by pivoting on $A[r] = A[\pi_j]$ is

$$P(\text{As or less balanced splits}) = \sum_{k \in S} \frac{1}{n}$$

Now note

$$|S| \approx 2(\alpha(n-1)+1)$$

since

$$|\{1,\cdots,\min\{j,n-(j-1)\}\}| = |\{1,\cdots,\min\{j,n-(j-1)\}\}| \approx \alpha(n-1)+1$$

Thus

$$\begin{aligned}
P(\text{As or less balanced splits}) &= \sum_{k\in S}\frac{1}{n} \\
&\approx 2(\alpha(n-1)+1)\frac{1}{n} \\
&= 2\alpha - 2\alpha/n + 2/n \\
&\approx 2\alpha
\end{aligned}$$

Since $2\alpha - 2\alpha/n + 2/n \approx 2\alpha$ for large n,
Hence

$$P(\text{More balanced splits}) = 1 - P(\text{As or less balanced splits}) \approx 1 - 2\alpha$$

4 Maximum of $q^2 + (n-q-1)^2$

We show $q^2 + (n-q-1)^2$ achieves a maximum over $q = 0, 1, \cdots, n-1$ when $q = 0$ or $q = n-1$.
First, note

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}q}[q^2 + (n-q-1)^2] &= 2q + 2(n-q-1)(-1) \\
&= 2q - 2n + 2q + 2 \\
&= 4q - 2n + 2
\end{aligned}$$

Then setting this first derivative to zero yields

$$\begin{aligned}
4q - 2n + 2 &= 0 \\
\implies \qquad q &= 1/2(n-1)
\end{aligned}$$

Next, we check the second derivative at this critical point:

$$\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}^2 q}[q^2 + (n-q-1)^2] &= \frac{\mathrm{d}}{\mathrm{d}q}[4q - 2n + 2] \\
&= 4 > 0
\end{aligned}$$

Since the second derivative is positive this is a minimum. Thus, we are left checking the boundaries.

When $q = 0$

$$q^2 + (n - q - 1)^2 = 0^2 + (n - 0 - 1)^2 = (n - 1)^2$$

When $q = (n - 1)$

$$q^2 + (n - q - 1)^2 = (n - 1)^2 + (n - (n - 1) - 1)^2 = (n - 1)^2$$

Also,

$$4q - 2n + 2\Big|_{q=0} = -2n + 2 < 0 \qquad \forall n > 1$$

so this function is decreasing at $q = 0$, making this boundary a local maximum.

$$4q - 2n + 2\Big|_{q=n-1} = 4n - 4 - 2n + 2 = 2n - 2 > 0 \qquad \forall n > 1$$

so this function is increasing at $q = n - 1$, making this boundary a local maximum. Thus, the maximum is achieved at $q = 0$ and $q = 1$.

5 Quicksort's best-case running time is $\Omega(n \log n)$

Let $T(n)$ be the best-case running time for quicksort on an array A of size n. Then the recurrence is

$$T(n) = \min_{0 \le q \le n-1} [T(q) + T(n - q - 1)] + \Theta(n)$$

We guess $T(q) \ge cq \log q$ for all $q < n$ (i.e. this is our strong inductive hypothesis). Then

$$T(n) = \min_{0 \le q \le n-1} [cq \log q + c(n - q - 1) \log q] + \Theta(n)$$

Now we find the minimum over $0 \le q \le n - 1$ by setting the first derivative to zero and solving for q.

$$\begin{aligned}
0 &= \frac{\mathrm{d}}{\mathrm{d}q} [cq \log q + c(n - q - 1) \log q] \\
&= \left[q\frac{1}{q} + \log q + \left(-\frac{n - q - 1}{n - q - 1} - \log(n - q - 1) \right) \right] \\
&= 1 + \log q - 1 - \log(n - q - 1) \\
&= \log q - \log(n - q - 1) \\
\implies \qquad 2^0 &= 2^{\log q - \log(n - q - 1)} \\
1 &= \frac{q}{n - q - 1} \\
(n - q - 1) &= q \\
q &= 1/2(n - 1)
\end{aligned}$$

Next, we check the second derivative:

$$
\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}^2 q}\left[cq\log q + c(n-q-1)\log q\right] &= \frac{\mathrm{d}}{\mathrm{d}q}\left[\log q - \log(n-q-1)\right] \\
&= \frac{1}{q} - \frac{1}{n-q-1} \\
\implies \qquad \left.\frac{\mathrm{d}^2}{\mathrm{d}^2 q}\right|_{q=1/2(n-1)} &= \frac{1}{1/2(n-1)} - \frac{1}{n-(1/2(n-1))-1} \\
&= \frac{1}{1/2(n-1)} - \frac{1}{1/2(n-1)} \\
&= 0
\end{aligned}
$$

Since the second derivative is zero, we consider the first derivative at $q = 1/2(n-1) \pm \epsilon$:

$$
\log\left(1/2(n-1)+\epsilon\right) + \log\left(n - (1/2(n-1)+\epsilon) - 1\right) = \log\left(1/2(n-1)+\epsilon\right) + \log\left(1/2(n-1)-\epsilon\right)
$$

Which is negative when $\epsilon < 0$, and positive when $\epsilon > 0$. Hence the objective function is decreasing for $q < 1/2(n-1)$ and increasing for $q > 1/2(n-1)$, so the function achieves a minimum at $q = 1/2(n-1)$.
Next, substituting into our original expression for $T(n)$ yields

$$
\begin{aligned}
T(n) &= [c \cdot q_{\text{min}} \log(q_{\text{min}}) + c \cdot (n - q_{\text{min}} - 1)\log(n - q_{\text{min}} - 1)] + \Theta(n) \\
&= [c \cdot (1/2(n-1))\log(1/2(n-1)) + c \cdot (n - (1/2(n-1)) - 1)\log(n - (1/2(n-1)) - 1)] + \Theta(n) \\
&= [c \cdot (1/2(n-1))\log(1/2(n-1)) + c \cdot (1/2(n-1))\log(1/2(n-1))] + \Theta(n) \\
&= [2c \cdot (1/2(n-1))\log(1/2(n-1))] + \Theta(n) \\
&= [c \cdot (n-1)\log(1/2(n-1))] + \Theta(n) \\
&= [c \cdot (n-1)\left(\log(n-1) - \log(2)\right)] + \Theta(n) \\
&= [c \cdot (n-1)\left(\log(n-1) - 1\right)] + \Theta(n) \\
&= c \cdot (n-1)\log(n-1) - c \cdot (n-1) + \Theta(n) \\
&= c \cdot (n-1)\log(n(1-1/n)) - c \cdot (n-1) + \Theta(n) \\
&= c \cdot (n-1)\left(\log(n) + \log(1-1/n)\right) - c \cdot (n-1) + \Theta(n) \\
&= c \cdot n\log(n) - c\log(n) + c \cdot n\log(1-1/n) - c\log(1-1/n) - c \cdot (n-1) + \Theta(n)
\end{aligned}
$$

Thus, to show

$$
T(n) > c \cdot n\log(n)
$$

we need to show that there exists some c such that for all $n > n_0$

$$
-c\log(n) + c \cdot n\log(1-1/n) - c\log(1-1/n) - c \cdot (n-1) + \Theta(n) < 0
$$

Well,

$$
\begin{aligned}
&-c\log(n) + c \cdot n \log(1 - 1/n) - c\log(1 - 1/n) - c \cdot (n-1) + \Theta(n) \\
&\quad = c(1 - \log(1-1/n) - \log(n)) + c\left(\log(1 - 1/n) - 1\right) n + \Theta(n)
\end{aligned}
$$

Next, note $c(1 - \log(1 - 1/n) - \log(n)) < 0$ for any reasonably large n (i.e. greater than $n = 4$), so if

$$c\left(\log(1 - 1/n) - 1\right) n + \Theta(n) < 0$$

then clearly

$$c(1 - \log(1-1/n) - \log(n)) + c\left(\log(1 - 1/n) - 1\right) n + \Theta(n) < 0$$

Similarly, note

$$c\left(\log(1 - 1/n) - 1\right) < -c$$

so again if

$$-cn + \Theta(n) < 0$$

then

$$c\left(\log(1 - 1/n) - 1\right) n + \Theta(n) < 0$$

Thus all we need to do is find c such that

$$-cn + \Theta(n) < 0$$

But that is obviously possible- simply take $c = 2 \cdot c_1$, where c_1 is the constant hidden in the lower bound implied by $\Theta(n)$.
Thus, we have found a c such that for all $n > n_0$,

$$-c\log(n) + c \cdot n \log(1 - 1/n) - c\log(1 - 1/n) - c \cdot (n-1) + \Theta(n) < 0$$

Hence

$$T(n) > c \cdot n \log(n)$$