Answer Set 2

Kazi Samin Mubasshir
Email: kmubassh@purdue.edu
PUID: 34674350

Problem 1

Contributors: Farhanaz Farheen, Tania Chakraborty, Devin Atilla Ersoy

Here, n is the size of the input array. Let $T(n)$ be the expected running time of the Randomized-Quicksort on inputs of size n. By convention, let $T(0) = 0$. Here, x_i is the pivot. The size of the left sub-array after partitioning is the rank of $x_i - 1$.

Making a hypothesis: We claim that the expected running time is at most $cn\log(n)$ for all $n \geq 1$. We prove this by induction on n. Let a be a constant such that partitioning of a size n subarray requires at most an steps.

For the base case, we can choose a value of c so that the claim hold.

For the induction step,let $n > 3$ and suppose that the claim holds for all values of n less than the current one.

The expected running time satisfies the following

$$T(n) \leq an + \sum_{i=0}^{n-1}(T(i) + T(n-1-i))\frac{1 - \frac{1}{(n+i)^2}}{\sum_{k=1}^{n} 1 - \frac{1}{(n+k)^2}}$$

$$= an + (T(0) + T(n-1))\frac{1 - \frac{1}{(n+1)^2}}{\sum_{k=1}^{n} 1 - \frac{1}{(n+k)^2}} + \ldots + (T(n-1) + T(0))\frac{1 - \frac{1}{(n+n)^2}}{\sum_{k=1}^{n} 1 - \frac{1}{(n+k)^2}}$$

$$= an + \frac{(T(0) + T(n-1))(2 - \frac{1}{(n+1)^2}) - \ldots - \frac{1}{(n+n)^2}}{\sum_{k=1}^{n} 1 - \frac{1}{(n+k)^2}}$$

The denominator converges to n for larger values of n

Hence,

$$T(n) \leq an + \frac{2}{n}(T(0) + T(n-1)) + \frac{2}{n}(T(1) + T(n-2)) + \ldots$$

$$T(n) \leq an + \frac{2}{n}\sum_{k=1}^{n} T(k)$$

By our induction hypothesis, this is at most

$$an + \frac{2c}{n}\sum_{k=1}^{n-1} k \cdot \log(k)$$

We know,

$$\frac{2c}{n}\sum_{k=1}^{n-1} k \cdot \log(k) \leq \int_{1}^{n} x\log(x)\,dx$$

The integration is equal to

$$\frac{1}{2}n^2\log(n) - \frac{n^2}{4} + \frac{1}{4}$$

This is at most,

$$\frac{1}{2}n^2\log(n) - \frac{n^2}{8}$$

By plugging this bound we have,

$$T(n) \leq cn\log(n) + (a - \frac{c}{4})n$$

Choosing $c > 4a$, Then

$$T(n) \leq cn\log(n)$$

Thus we have proven an upper bound of $O(n\log(n))$ comparisons in expectation for this variant of QuickSort.

Problem 2

Contributor: Farhanaz Farheen

We found that

- For $k = 1$, $\Theta(n^2)$
- For $k = 2$, $\Theta(n^{\frac{4}{3}})$
- For $k = 3$, $\Theta(n^{\frac{8}{7}})$

If we work it out for $k = 4$

Upper Bound: Lets divide the elements into groups of k [TBD] and find the maximum element in each group.

- In round 1 can obtain the maximum y_i for each group i. We get $k \cdot \frac{n^2}{k^2} = \frac{n^2}{k}$ comparisons in round 1.
- By 2 round protocol, the maximum of the y_is can be obtained using $2 \cdot k^{\frac{4}{3}}$ comparisons in the remaining three rounds.
- By 3 round protocol, the maximum of the y_is can be obtained using $3 \cdot k^{\frac{8}{7}}$ comparisons in the remaining two rounds.

The total number of comparisons for $round = 4$ is $\leq \frac{n^2}{k} + 2 \cdot k^{\frac{4}{3}} + 3 \cdot k^{\frac{8}{7}}$.
By setting $k = n^{\frac{14}{15}}$ we get at most $\frac{n^2}{k} + 2 \cdot k^{\frac{4}{3}} + 3 \cdot k^{\frac{8}{7}} = 6 \cdot n^{\frac{16}{15}} \in O(n^{\frac{16}{15}})$

We can see a pattern here. For each round, the upper bound is $n^{\frac{2^{round\_no}}{2^{round\_no}-1}}$

So we can say that $f(n,k) = O(n^{\frac{2^k}{2^k-1}})$

Lower Bound: Use same technique as for r=3 rounds:

- assume the first round uses at most k' comparisons (if it uses less, we are done).
- Find large independent set S (By Lemma 1, can find of size $\geq \frac{n^2}{8 \cdot k'}$)
- Thus any$k$ round protocol uses either k' comparisons in the first round or $\geq (\frac{n^2}{8 \cdot k'})^{\frac{2^{k-1}}{2^{k-1}-1}}$ in the next rounds.
- By setting $k' = n^{\frac{2^k}{2^k-1}}$, we get a lower bound of $\Omega(n^{\frac{2^k}{2^k-1}})$

So $f(n,k) = \Theta(n^{\frac{2^k}{2^k-1}})$

Problem 3

Observations:

- Mergesort performed better. Quicksort was also close, randomly selecting the pivot made it's runtime closer to $O(n \log(n)$
- Merge Sort made fewer comparisons.