CMPT 210: Probability and Computation

Lecture 20

Sharan Vaswani

July 22, 2022

Assignment 4 released. Due on August 2.

Randomized QuickSort

Given an array A of n distinct numbers, sort the elements in A in increasing order.

Algorithm Randomized QuickSort

```
function QuickSort(A)
If Length(A) = 1, return A.
Select p ∈ A uniformly at random.
Construct arrays Left := [x ∈ A|x < p] and Right := [x ∈ A|x > p].
Return the concatenation [QuickSort(Left), p, QuickSort(Right)].
```

Randomized QuickSort

If $A = [2, 7, 0, 1, 3]$ and according to the algorithm, $p \sim \text{Uniform}(A)$. Say $p = 3$. For this step, Left $= [2, 0, 1]$ and Right $= [7]$.

The algorithm will return the concatenation $[\text{QuickSort}([2, 0, 1]), 3, \text{QuickSort}([7])] = [\text{QuickSort}([2, 0, 1]), 3, 7]$.

Total number of comparisons $= 4$ (comparing every element to the pivot $= 3$.)

In the second step, for running the algorithm on $[2, 0, 1]$, say $p = 1$. For this step, Left $= [0]$ and Right $= [2]$ and the algorithm will return the concatenation $[\text{QuickSort}([0]), 1, \text{QuickSort}([2]), 3, 7] = [0, 1, 2, 3, 7]$.

Total number of comparisons $= 4$ (from step 1) $+ 2$ (comparing elements in Left to pivot $= 1$.)

Q: Run the algorithm if $p = 2$ in the first step?

Ans: Left $= [0, 1]$ and Right $= [7, 3]$. Running the algorithm on $[0, 1]$ will return $[0, 1]$ and on $[7, 3]$ will return $[3, 7]$. Hence the algorithm will return the concatenation $[0, 1, 2, 3, 7]$ thus sorting the array.

Questions?

Randomized QuickSort

Claim: For a set A with n distinct elements, the expected (over the randomness in the pivot selection) number of comparisons for QuickSort is $O(n \ln(n))$.

Let us write the elements of A in sorted order, $a_1 < a_2 < \ldots < a_n$. Let X be the r.v. equal to the number of comparisons performed by the algorithm.

Observation: Every pair of elements is compared at most once since we do not include the pivot in the recursion.

For $i < j$, let $E_{i,j}$ be the event that elements i and j are compared, and define $X_{i,j}$ to be the indicator r.v. equal to 1 if event $E_{i,j}$ happens. Hence, $X = \sum_{1 \leq i < j \leq n} X_{i,j}$, and

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{1 \leq i < j \leq n} X_{i,j}\right] = \sum_{1 \leq i < j \leq n} \mathbb{E}[X_{i,j}] = \sum_{1 \leq i < j \leq n} \Pr[E_{i,j}] \qquad \text{(Linearity of expectation)}$$

Randomized QuickSort

Fix $i < j$ (meaning that $a_i < a_j$) and let $R = [a_i, \dots, a_j]$.

Claim: $E_{i,j}$ happens if and only if the first pivot selected from R is either a_i or a_j.

Elements a_i and a_j are compared if they are still in the same sub-problem at the time that one of them is chosen as the pivot. Elements a_i and a_j are split into different recursive sub-problems at precisely the time that the first pivot is selected from R. If this pivot is either a_i or a_j, then they will be compared; otherwise, they will not.

In our example, $A = [2, 7, 0, 1, 3]$ and suppose $a_i = 0$ and $a_j = 2$. After $p = 3$ is chosen, Left $= [2, 0, 1]$. Both 0 and 2 are compared to the pivot $p = 3$, and end up in the same sub-problem. Hence the elements in $R = [0, 1, 2]$ appear together.

For the next step, when recursing on Left, if $p = 1$, then Left $= [0]$ and Right $= [2]$ and elements 0 and 2 will never be compared. On the other hand, if $p = 2$, then since each element is compared to the pivot, 0 and 2 will be compared.

Hence, $E_{i,j}$ will happen if the first pivot selected from R is either a_i or a_j.

Claim: $\Pr[a_i$ or a_j is the first pivot selected from $R] = \frac{2}{|R|} = \frac{2}{j-i+1}$.

In our example, if $a_i = 0$ and $a_j = 2$ and say $p = 7$, then after the first step, Left $= [2, 0, 1, 3]$. Hence the elements in $R = [0, 1, 2]$ appear together in the same sub-problem.

For the second step, when recursing on $T = [2, 0, 1, 3]$, since p is chosen uniformly at random, conditioned on the event that $p \in R$, p is also uniformly random on R. Formally, for $x \in T$, $\Pr[p = x] = \frac{1}{|T|}$.

$$\Pr[p = x | p \in R] = \frac{\Pr[p = x \cap p \in R]}{\Pr[p \in R]} = \frac{\Pr[p = x]}{\Pr[p \in R]} \quad (\text{For all } x \notin R,\ \Pr[p = x \cap p \in R] = 0)$$

$$= \frac{1/|T|}{\sum_{x \in R} \Pr[p = x]} = \frac{1/|T|}{|R|/|T|} = \frac{1}{|R|}$$

Hence, the probability of selecting either 0 or 2 (a_i and a_j respectively) in a sub-array (T in the above example) that contains R ($[0, 1, 2]$ in the example) is $2/|R| = 2/(j - i + 1)$ (equal to $2/3$ in the example).

Randomized QuickSort

Putting everything together, $\Pr[E_{i,j}] = \frac{2}{j-i+1}$.

Hence, the expected number of comparisons is equal to

$$\mathbb{E}[X] = \sum_{1 \le i < j \le n} \frac{2}{j-i+1} = \sum_{i=1}^{n-1} \left[\sum_{j=i+1}^{n} \frac{2}{j-i+1} \right] = 2 \sum_{i=1}^{n-1} \left[\frac{1}{2} + \frac{1}{3} \dots + \frac{1}{n-i+1} \right]$$

$$< 2 \sum_{i=1}^{n-1} \left[\frac{1}{2} + \frac{1}{3} \dots + \frac{1}{n} \right] < 2n \left[\frac{1}{2} + \frac{1}{3} \dots + \frac{1}{n} \right]$$

$$\le 2n \int_1^n \frac{dx}{x} = 2n \ln(n) \qquad \text{(Bounding the harmonic series similar to Lecture 14)}$$

Hence, the expected number of comparisons required for Randomized QuickSort is $O(n \ln(n))$.

Q: What is the number of comparisons for Randomized QuickSort in the worst-case?

Similar to Randomized QuickSelect, for Randomized QuickSort, the worst-case happens when the pivot is selected to be the minimum (or maximum) element in the sub-array in each iteration. And hence the worst-case complexity is $O(n^2)$.

Questions?