CSE515 Advanced Algorithms
Notes on Lecture 24: Random Binary Search Trees

Antoine Vigneron

May 19, 2021

1 Proof of correctness of RandomPermutation

In the following, we prove that RandomPermutation produces a permutation of the input chosen uniformly at random.

Without loss of generality, assume that the input is $A[1 \ldots n] = (1, 2, \ldots, n)$, so we want to prove that the output is a permutation of $(1, \ldots, n)$ chosen uniformly at random.

At the ith iteration of the loop, the algorithm generates a random number n_i between i and n. So the output is determined by the n-tuple of integers $(n_1, n_2, \ldots, n_n)$ where $n_i \in i, \ldots, n$. There are $n \times (n-1) \times \cdots \times 2 \times 1 = n!$ such tuples, which are equally likely. We will prove that each permutation corresponds to exactly one such tuple, and thus each permutation has probability exactly $1/n!$ of being computed by this algorithm.

We first illustrate this fact by an example. For instance, suppose that we start from $[1, 2, 3, 4, 5]$ and the output permutation is $[3, 2, 4, 1, 5]$. Then necessarily $n_1 = 3$, and after the first swapping the array was $[3, 2, 1, 4, 5]$. Then in order to obtain 2 at the second position, we must have $n_2 = 2$. After swapping 2 with itself, the array is still $[3, 2, 1, 4, 5]$. As 4 appears in 3rd position, $n_3 = 4$, and after swapping 1 with 4, we obtain the array $[3, 2, 4, 1, 5]$. After this, we must have $n_4 = 4$ and $n_5 = 5$ to keep the last two elements in order. So the only 5-tuple that generates $[3, 2, 4, 1, 5]$ is $(n_1, \ldots, n_5) = (3, 2, 4, 4, 5)$.

This generalizes to any value of n and any output permutation $\sigma = (\sigma_1, \ldots, \sigma_n)$. We must have $n_1 = \sigma_1$, then n_2 is the index of σ_2 in the array obtained after swapping $A[1]$ with $A[n_1]$, and n_3 is the index of σ_3 in the array obtained after swapping $A[2]$ with $A[n_2]$ ...

This shows that each of the $n!$ possible permutation is obtained from exactly one of the $n!$ possible tuple, hence each permutation is generated with probability $1/n!$.

2 Analysis of ConstructRandomBST

Let T be a random BST with n nodes. We want to prove that the expected value $E[P]$ of its total path length $P = P(T)$ is $O(n \log n)$.

We say that a node $v \in T$ is *central* if the left and right subtrees rooted at T each have at most $3|T_v|/4$ nodes, where $|T_v|$ is the size of the subtree rooted at v. It means that the key of v is between the $n/4$th and the $3n/4$th smallest key in T_v, and thus v is central with probability $1/2$. (To be more precise, it can be slightly larger than $1/2$, but we will ignore this issue as we did in the previous two lectures.)

We say that a node v is in phase j if the size of the subtree T_v rooted at v satisfies

$$\left(\frac{3}{4}\right)^{j+1} n < |T_v| \leqslant \left(\frac{3}{4}\right)^{j} n.$$

By this definition, we have $n(3/4)^j \geqslant 1$ so $j \leqslant \log_{4/3} n$. Observe that if v is central, then its children are in phase at least $j+1$.

We now consider an arbitrary node u, and we want to bound its expected depth. Let P_u be the path from the root to u. When we insert u, we traverse this path P_u. Each time we go down one node, we move to a smaller subtree. As the root of this subtree was picked at random by our algorithm, the probability that it is central is $1/2$, independently of the previous nodes in the path. Hence, when we travel along P_u from the root, each node is central with probability $1/2$.

How many nodes along P_u are in phase j? Suppose $v \in P_u$ is in phase j. As we travel down, if we encounter a central node, then the phase increases. As a node is central with probability $1/2$, the waiting time bound tells us that the expected number of nodes before we find the next central node is 2. Therefore, the expected number of nodes in phase j along P_u is at most 2.

As the number of phases is at most $\log_{4/3} n$, the expected length of the path P_u from the root to v is at most $2\log_{4/3} n$. By linearity of expectation, it implies that

$$\begin{aligned} E[P] &\leqslant 2n \log_{4/3} n \\ &= O(n \log n). \end{aligned}$$