COMP690 Fall 2020 Homework 2

Names: Andrew Cheng, Jad Hamdan

Exercise 3.39

Define $A_n = \sum_{i<j} \mathbf{1}_{[\sigma_i < \sigma_j]}$. We can determine the first moment in two different ways - by using combinatorics or by using the properties of random binary search trees.

Approach 1:

$$\begin{aligned}
\mathbb{E}(A_n) &= \sum_{i<j} \mathbb{E}\{\mathbf{1}_{[\sigma_i<\sigma_j}\} \\
&= \sum_{i<j} \mathbb{P}\{\sigma_i < \sigma_j\} \\
&= \sum_{i<j} \frac{\text{\# ways to pick (i,j) s.t} i < j}{\text{total \# ways to pick ordered pairs}} \\
&= \sum_{i<j} \frac{\binom{n}{2}}{2! \times \binom{n}{2}} \\
&= \frac{1}{2} \sum_{i<j} 1 \\
&= \frac{n(n-1)}{4}
\end{aligned}$$

Approach 2: We build n random binary search trees in n iterations in the following manner:

1. At iteration i, use σ_i as the root and insert the elements $\sigma_{i+1}, \dots, \sigma_n$ via standard insertion.

Repeat the above for $i \in \{1, \dots, n\}$ to build n random binary search trees. Define the random variable B_i to be the size of the right subtree built at iteration i. Note that $B_i = \sum_{i<j} \mathbf{1}_{[\sigma_i<\sigma_j]}$

$$\begin{aligned}
\mathbb{E}\{A_n\} &= \mathbb{E}\{\sum_{j=1}^{n} B_i\} \\
&= \sum_{i=1}^{n} \mathbb{E}\{B_i\} \\
&= \sum_{i=1}^{n} \frac{n-i}{2} \\
&= \frac{n(n-1)}{4}
\end{aligned}$$

where the second last equality comes from the fact that the we expect a random binary search tree to be balanced, i.e., the expected size of the right subtree is half the size of the

tree excluding the root.
Now we compute the second moment.

$$\begin{aligned}\mathbb{E}\{A_n^2\} &= \mathbb{E}\Bigg\{\bigg(\sum_{i<j}\mathbf{1}_{[\sigma_i<\sigma_j]}\bigg)^2\Bigg\} \\ &= \sum_{(i<j),(k<l)}\mathbb{E}\Big\{\mathbf{1}_{[\sigma_i<\sigma_j]}\mathbf{1}_{[\sigma_k<\sigma_l]}\Big\}\end{aligned}$$

The term can be split into 5 cases:

1. the pairs (i,j) and (k,l) are identical, where the product of the indicator terms becomes 1/2 ($\binom{n}{2}(1/2)$ such terms)
2. the pairs (i,j) and (k,l) have no elements in common, where the product of the indicator terms becomes 1/4 ($\binom{n}{4}\binom{4}{2}$ such terms)
3. $[i=k]$ and $[k<l \text{ or } l<k]$, where the product becomes 1/3 ($\binom{n}{3} * 2$ such terms)
4. $[j=l]$ and $[i<k \text{ or } k>i]$ (symmetric to the previous case)
5. $j=k$, with probability 1/6, where the product becomes 1/6 ($\binom{n}{3} * 2$ such terms)

This yields

$$\mathbb{E}\{A_n^2\} = \binom{n}{2}(1/2)(1/2) + \binom{n}{4}\binom{4}{2}(1/4) + \binom{n}{3}4(1/3) + \binom{n}{3}2(1/6)$$

Using Wolfram Alpha, we see that

$$\lim_{n\to\infty} \operatorname{Var} A_n/(n^3/36) = 1 \tag{1}$$

Thus $\operatorname{Var}(A_n) \sim cn^3$ where $c = 1/36$. We can show $\frac{A_n}{E\{A_n\}} \to 1$ in probability by using Chebyshev. Let $\epsilon > 0$

$$\begin{aligned}\mathbb{P}\bigg(\big|\frac{A_n}{\mathbb{E}(A_n)} - 1\big| \geq \epsilon\bigg) &= \mathbb{P}\bigg(\big|A_n - \frac{n(n-1)}{4}\big| \geq \frac{\epsilon \times n(n-1)}{4}\bigg) \\ &\leq \frac{\operatorname{Var}(A_n)}{(\frac{\epsilon\times n(n-1)}{4})^2} \\ &\xrightarrow{\text{as n goes to infinity}} 0 \textit{(Since the variance is cubic and the denominator of order 4)}\end{aligned}$$

Thus we have our result. $\square$

Exercise 3.40

Observe that the artifical nodes of the complete binary tree induces a partition of n equal length intervals on the set $[0,1]$. Each datapoint $U_i \overset{iid}{\sim} Unif[0,1]$ falls into one of the n intervals with equal probability. Define the random variable N_i as the number of keys that fall into the i^{th} interval. Now notice that H_n is upperbounded by $\max\{N_1, \ldots, N_n\} + h$, where $N_1 + \cdots + N_n = n$. The quantity $\mathbb{E}\{max(N_1, \ldots, N_n)\} = o(log_2(n))$ is a well-known result from the "balls into bins" problem, a popular problem in computer science (Prof. Devroye said we could use this result without proof). So we have showed that

$$H_n \leq \underbrace{h}_{\log_2 n} + max(N_1, \ldots, N_n)$$

$$\begin{aligned} \implies E(H_n) &\leq E\Bigg(\log_2(n) + max(N_1, \ldots, N_n)\Bigg) \\ &= \log_2 n + \underbrace{E\Bigg((N_1, \ldots, N_n)\Bigg)}_{o(\log_2 n)} \end{aligned}$$

But we trivially have that $E(H_n) \geq log_2(n)$. Thus we yield our result $E(H_n) \sim \log_2 n$ $\square$

Exercise 3.41

In approach 1 we will show the results using the direct formula give in notes. We will also provide a derivation of the formula.

Approach 1: As seen in lectures

$$P(D_n = k) = \frac{1}{n!} \begin{bmatrix} n-1 \\ k \end{bmatrix} 2^k \quad 1 \leq k \leq n \tag{2}$$

where $[\cdot]$ is the (signless) Stirling number of the first kind. Thus we have $Q_{n,k} = \begin{bmatrix} n-1 \\ k \end{bmatrix} 2^k$.

Let c(n,k) be the *signless* Stirling number of the first kind i.e. $c(n,k) = \begin{bmatrix} n \\ k \end{bmatrix}$.

Let's show $c(n-1,k)2^k \in \mathbb{Z}$. The base cases to compute the signless Stirling number of the first kind is as follows:

$$\begin{aligned} &c(n,n) = 1 \\ &c(0,0) = 1 \\ &c(n,0) = 0 \quad \forall n \geq 1 \end{aligned}$$

and the recurrence relation is

$$c(n,k) = c(n-1,k-1) + (n-1) \times c(n-1,k) \tag{3}$$

Computing $c(n-1,k)$ requires multiplying and adding integers. By $\mathbb{Z}$ closed under addition and subtraction we have $c(n-1,k) \in \mathbb{Z}$. Clearly $2^k \in \mathbb{Z}$ for $k \in \mathbb{Z}$. Then we have

$$|Q_{n,k}| = c(n-1,k)2^k \in \mathbb{Z}$$
$$\implies Q_{n,k} \in \mathbb{Z}$$

We can solve this problem in $O(n^2)$ using dynamic programming. Compute $c(n,k)$ using the base cases and recurrence relation mentioned above via dynamic programming in $O(n^2)$. We can compute 2^k in $O(\log k)$. We output: $2^k c(n-1,k)$.

Deriving Approach 1

We can convince ourselves that $Q_{n,k} := n!\mathbb{P}\{D_n = k\}$ is integer valued without the formula above. This is because the event $\{D_n = k\}$ can be thought of as a condition on permutations of n elements (we're seeking those for which the corresponding cartesian tree satisfies this condition). The total number of such permutations being $n!$, $\mathbb{P}\{D_n = k\}$ will thus be of the form $k/n!$ for some integer k, and, consequently, $Q_{n,k} = n! * (k/n!) = k$ is integer valued.

Now, let us derive the formula. Let D_n be the depth of σ_n the node with the largest time stamp in a random binary search tree on n nodes. Notice that $P\{D_n = k\}$ depends on two events: the first is when σ_n lies directly below the previously inserted node σ_{n-1} and the second is when σ_n is inserted under a node inserted before σ_{n-1}.

The probability of the former event occurring can be seen as the probability of D_{n-1} taking depth *k-1*, but there are only two out of n "slots" to fall under σ_{n-1}. Thus the probability of the first event is $\frac{2}{n}P\{D_{n-1} = k-1\}$. We can invoke the mirroring argument for the latter case: if the σ_n falls under a node inserted before σ_{n-1} then the following distributions are identical $(\sigma_1, \ldots, \sigma_{n-1}, \sigma_n) \overset{\mathcal{L}}{=} (\sigma_1, \ldots, \sigma_n, \sigma_{n-1})$. Thus we have $P\{D_n = k\} = P\{D_{n-1} = k\}$. But σ_{n-1} had *(n-1)* out of n slots to fall in. Thus the second case yields the probability $\frac{n}{n-1}P\{D_{n-1} = k\}$. This yields the recurrence relation:

$$P\{D_n = k\} = \frac{2}{n}P\{D_{n-} = k-1\} + \frac{(n-1)}{n}P\{D_{n-1} = k\} \tag{4}$$

Thus we can express this as a recursion with the following:

$$\begin{aligned} Q_{n,k} &= n!\mathbb{P}(D_n = k) \\ &= n!\left(\frac{n-1}{n}\mathbb{P}\{D_{n-1} = k\} + \frac{2}{n}\mathbb{P}\{D_{n-1} = k-1\}\right) \\ &= (n-1)Q_{n-1,k} + 2Q_{n-1,k-1} \end{aligned} \tag{5}$$

which yields the formula in approach 1.

Exercise 3.42

To begin with, we need to find the probability of a single terminal occurrence of C_h given a "window" of nodes. To illustrate our approach, let us examine terminal occurrences of C_0 and C_1.

A terminal occurrence of C_0 is simply a leaf. As seen in class, a node is a leaf if its timestamp τ_i (using a Cartesian tree model) is the largest of $\tau_{i-1}, \tau_i, \tau_i$. This event thus occurs on a window of 3 nodes, and can be viewed as a condition on permutations of 3 elements: τ_i is a leaf if and only if the permutation of $\tau_i, \tau_i, \tau_{i+1}$'s ranks is $[1 \quad 3 \quad 2]$ or $[1 \quad 3 \quad 2]$. Only two out of six (3!) permutations satisfy this condition, giving us a probability of 1/3. In the following paragraphs, we denote p_h the probability of a terminal occurrence of C_h on a window of size $2^{h+1}+1$ (thus $p_0 = 1/3$).

We're now interested in permutations of 5 elements to compute p_1. We want the second and penultimate elements in this permutation to be timestamps of leaves, hence they need to be "local" maxima. Furthermore, we need the center element to be 3, since our leaves could be children of the first/last node of our window otherwise. Only four permutations satisfy these conditions: $[1 \quad 4 \quad 3 \quad 5 \quad 2]$, $[2 \quad 4 \quad 3 \quad 5 \quad 1]$, $[1 \quad 5 \quad 3 \quad 4 \quad 2]$, $[2 \quad 5 \quad 3 \quad 4 \quad 1]$. We can conclude that $p_1 = 4/5! = 1/30$.

We can reformulate this C_1 argument in a way that mainly concerns itself with so-called "middle elements". More specifically, we begin by selecting the first and last elements of the permutation (probability $2 * \frac{1}{5}\frac{1}{4}$, since we either pick 1 as first and 2 as last or vice-versa). Now, we only have a single choice for the middle element of this window of size 5, and that's 3, meaning we have a probability of $\frac{1}{5-2} = \frac{1}{3}$ of getting this right. Now we're left with two holes, each of size 1, that are correctly filled with probability 1.

While this reformulation may seem cumbersome, or unnecessary, it has the benefit of generalizing nicely to any C_h, by repeatedly taking the probability of picking an adequate "middle element". Letting $w = 2^{h+1}+1$ be our window size, we pick our first and last elements with probability $2 \cdot \frac{1}{2^{h+1}+1} \cdot \frac{1}{2^{h+1}}$. We then have a single choice for our middle element (3), picked with probability $\frac{1}{2^{h+1}-1}$. This leaves us with two holes of width $2^h - 1$ each, for which we have a probability of $\frac{1}{2^h-1}$ of picking a valid middle element (respectively). The probability that both are correct is thus $\frac{1}{2^h-1}^2$. We now have $2^2 = 4$ holes of size $w^{h-1}-1$: applying the same line of reasoning gives us a probability of $\frac{1}{2^{h-1}-1}^4$ of picking correct middle elements in these holes. We repeat this process until all the remaining holes are of size 1 to get the probability of a valid permutation, which is p_h.

We can therefore conclude that p_h is given by the following product

$$p_h = 2 \cdot \frac{1}{2^{h+1}+1} \cdot \frac{1}{2^{h+1}} \cdot \prod_{i=1}^{h} \frac{1}{2^{i+1}-1}^{2^{h-i}}$$

The expected number of terminal occurrences of C_h is thus given by the sum of p_h over all possible windows. Let k_h be such that the number of windows is equal to $n - k_h$ (k_h that grows as h grows). We thus have

$$\mathbb{E}\{N_h\} = \sum_{i=1}^{n-k_h} p_h = (n-k_h)p_h$$

since p_h takes the same value on every window (we ignore the slight difference of so-called "edge windows", the first and last windows, as it doesn't affect the final limit).

We many now proceed to prove the main claim. We will do so in two parts.
Case: $h \geq (1+\epsilon)\log_2 \log n$
Lemma: $p_h < \frac{1}{e^{2^{h-1}}}$
Proof Let $D_h = 1/p_h$. It suffices to show that $\log(D_h) > 2^{h-1}$ and the lemma follows.

$$\begin{aligned}
\log(D_h) &= -\log(2) + \log(2^{h+1}+1) + \log(2^{h+1}) + \sum_{i=1}^{h} 2^{h-i}\log(2^{i+1}-1) \\
&\geq (2h+1)\log(2) + 2^h \sum_{i=1}^{h} \frac{\log(2^{i+1})-1}{2^i} \\
&\geq (2h+1)\log(2) + 2^h \left(\frac{2\log(2)-1}{2} + \sum_{i=2}^{h} \frac{\log(2^{i+1})-1}{2^i} \right) \\
&\geq (2h+1)\log(2) + 2^h \left(\frac{2\log(2)-1}{2} + \sum_{i=2}^{h} \frac{\log(2)}{2^i} \right) \\
&= (2h+1)\log(2) + 2^h \left(\frac{2\log(2)-1}{2} + \frac{\log(2)}{2} - \frac{\log(2)}{2^h} \right) \\
&\geq (2h+1)\log(2) + 2^h \left(\frac{2\log(2)-1}{2} + \frac{\log(2)}{2} \right) - \log(2) \\
&\geq (2h)\log(2) + 2^h/2 \\
&\geq 2^{h-1} \quad \square
\end{aligned}$$

We can apply this lemma in our computation of $\mathbb{E}\{N_h\}$ to get:

$$\begin{aligned}
\mathbb{E}\{N_h\} &= (n-k_h)p_h \\
&\leq np_h \\
&\leq np_{(1+\epsilon)\log_2 \log n} \quad \textit{(since } p_h \textit{ is decreasing)} \\
&\leq n\frac{1}{e^{2^{(1+\epsilon)\log_2 \log n - 1}}} \quad \textit{(by Lemma)} \\
&= n\frac{1}{\sqrt{n^{\log(n)^\epsilon}}} \quad \textit{(by logarithm rules)} \\
&\xrightarrow{\text{as n goes to infinity}} 0\square
\end{aligned}$$

This proves the first case. We may now proceed to the second case:
Case: $h \leq (1-\epsilon)\log_2 \log n$
Lemma: $p_h < \frac{1}{e^{2^{h+2}}}$

Proof Let $D_h = 1/p_h$. It suffices to show that $\log(D_h) < 2^{h+1}$ and the lemma follows.

$$\begin{aligned}
\log(D_h) &= -\log(2) + \log(2^{h+1}+1) + \log(2^{h+1}) + \sum_{i=1}^{h} 2^{h-i}\log(2^{i+1}-1) \\
&\leq 1 - \log(2) + (h+1)\log(2) + 2^h \log(2) \sum_{i=1}^{h} \frac{i+1}{2^i} \\
&\leq 1 + (h-1)\log(2) + 2^h \log(2) \left(\sum_{i=1}^{h} \frac{i}{2^i} + \sum_{i=1}^{h} \frac{1}{2^i} \right) \\
&\leq 1 + (h-1)\log(2) + 2^h \log(2) \left(2 + (2 - \frac{1}{2^h}) \right) \\
&\leq 2^h + 2^{h+2}\log(2) - \log(2) \\
&\leq 2^{h+2}(\frac{1}{4} + \log(2)) \\
&\leq 2^{h+2} \square
\end{aligned}$$

We can apply this lemma in our computation of $\mathbb{E}\{N_h\}$, once again, to show that the quantity explodes to infinity. Before that, however, a quick point needs to be made about k_h.

In a tree with n nodes, if we consider windows of size w, there are $n - w + 1$ possible windows. We add 2 to this number, to consider the edge cases on each side of the tree (cases where a terminal occurence of C_h's leftmost leaf would be the whole tree's leftmost leaf, and similarly for the right side). If we're hunting for terminal occurences of C_h, we use a window of size $2^{h+1}+1$, hence $k_h = 2^{h+1} + 1 - 3 = 2^{h+1} - 2$. Finally, since we're in the case where $h \leq (1-\epsilon)\log_2 \log(n)$, we have $k_h \leq 2\log(n)^{1-\epsilon} - 2 \leq \log(n)^{1-\epsilon}$. Using this alongside our second lemma yields:

$$\begin{aligned}
\mathbb{E}\{N_h\} &= (n - k_h)p_h \\
&\geq (n - 2\log(n)^{1-\epsilon})p_h \\
&\geq (n - 2\log(n)^{1-\epsilon})p_{(1-\epsilon)\log_2 \log n} \quad \textit{(since } p_h \textit{ is decreasing)} \\
&\geq (n - 2\log(n)^{1-\epsilon})\frac{1}{e^{2^{(1+\epsilon)\log_2 \log n + 2}}} \quad \textit{(by Lemma)} \\
&= (n - 2\log(n)^{1-\epsilon})\log(n)^{\epsilon 4} \quad \textit{(by logarithm rules)} \\
&= n\log(n)^{4\epsilon} - 2\log(n)^{3\epsilon} \\
&\xrightarrow{\text{as n goes to infinity}} \infty \square
\end{aligned}$$

Combining the two cases solves the main claim. Thanks for coming to my TED Talk.