CSOR W4231 Analysis of Algorithms - Spring 2021
Homework #4

Joseph High - `jph2185`

March 26, 2021

Problem 1

(a) At every time step $t > 1$, there are $t-1$ possible destinations for the outgoing edge leaving v_t in time step t. Since a destination node is chosen uniformly at random, the probability that node v_j is chosen in time step t is $\frac{1}{t-1}$.

Let X denote the random variable equal to the total number of edges entering node v_j at time step n. For $t > j$, let X_t denote the (random) indicator variable that indicates whether or not v_j is the destination of the edge leaving node v_t at time step t. That is,

$$X_t = \begin{cases} 1, & \text{if node } v_j \text{ is chosen as the destination node at time step } t, \text{ for } t > j \\ 0, & \text{otherwise} \end{cases}$$

Then $X = \sum_{t=j+1}^{n} X_t$ and $P\{X_t = 1\} = \frac{1}{t-1}$ in time step t. We then have:

$$\begin{aligned}
\mathbb{E}[X] = \mathbb{E}\left[\sum_{t=j+1}^{n} X_t\right] = \sum_{t=j+1}^{n} \mathbb{E}[X_t] &= \sum_{t=j+1}^{n} P\{X_t = 1\} \\
&= \sum_{t=j+1}^{n} \frac{1}{t-1} = \sum_{t=j}^{n-1} \frac{1}{t} \\
&= \sum_{t=1}^{n-1} \frac{1}{t} - \sum_{t=1}^{j-1} \frac{1}{t} \\
\textit{(n-1)}^{st} \textit{ and } (j-1)^{st} \textit{ harmonic numbers} \quad &= H_{n-1} - H_{j-1} \\
\textit{from equation A.7 in Appendix A.1} \quad &= [\ln(n-1) + O(1)] - [\ln(j-1) + O(1)] \\
&= \ln\left(\frac{n-1}{j-1}\right) + O(1)
\end{aligned}$$

Furthermore, the expected number of edges entering node v_j is bounded by $\Theta\left(\ln\left(\frac{n-1}{j-1}\right)\right) = \Theta\left(\ln\left(\frac{n}{j}\right)\right)$.

(b) Let X denote the random variable equal to the total number of nodes in G with no incoming edges at time step n. Let X_i denote the (random) indicator variable that indicates whether or not v_i has no incoming edges at time step i. That is,

$$X_i = \begin{cases} 1, & \text{if } \ indeg(v_i) = 0 \ \text{ at time step } n \\ 0, & \text{otherwise} \end{cases}$$

Then, $X = \sum_{i=1}^{n} X_i$ denotes the number of nodes with no incoming edges.

For the following, let (v_k, v_i) denote the directed edge leaving node v_k and entering v_i.

$$\begin{aligned}
\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i] &= \sum_{i=1}^{n} P\{X_i = 1\} \\
&= \sum_{i=1}^{n} P\left[\{(v_{i+1}, v_i) \notin E\} \cup \{(v_{i+2}, v_i) \notin E\} \cup \cdots \cup \{(v_n, v_i) \notin E\}\right] \\
&= \sum_{i=1}^{n} P\{(v_{i+1}, v_i) \notin E\} P\{(v_{i+2}, v_i) \notin E\} \cdots P\{(v_n, v_i) \notin E\} \\
&= \sum_{i=1}^{n} \left(1 - \frac{1}{i}\right)\left(1 - \frac{1}{i+1}\right)\left(1 - \frac{1}{i+2}\right) \cdots \left(1 - \frac{1}{n-1}\right) \\
&= \sum_{i=1}^{n} \left(\frac{i-1}{i}\right)\left(\frac{i}{i+1}\right)\left(\frac{i+1}{i+2}\right) \cdots \left(\frac{n-2}{n-1}\right) \\
&= \sum_{i=1}^{n} \frac{i-1}{n-1} \\
&= \frac{1}{n-1}\left[\sum_{i=1}^{n} i - \sum_{i=1}^{n} 1\right] \\
&= \frac{1}{n-1}\left[\frac{n(n+1)}{2} - n\right] = \frac{1}{n-1}\left[\frac{n(n-1)}{2}\right] \\
&= \frac{n}{2}
\end{aligned}$$

Hence, the expected number of nodes with no incoming edges is $\dfrac{n}{2}$.

Problem 2

Did not have time to complete. Move on to Problem 3.

Problem 3

W.l.o.g., let $\{b_1, b_2, b_3, \dots, b_n\}$ be any of the $n!$ possible permutations of the n bids, where index i for bid b_i indicates the order in which sellers place their bids. That is, bid b_1 is placed first, b_2 is placed second, and so on. Have the seller accept the first highest bid *after* bid $b_{\lceil \frac{n}{2} \rceil}$ has been placed. That is, for any $b_k \in \{b_{\lceil \frac{n}{2} \rceil+1}, b_{\lceil \frac{n}{2} \rceil+2}, \dots, b_n\}$ accept b_k only if $b_k \geq \max\{b_1, b_2, \dots, b_{\lceil \frac{n}{2} \rceil}\}$.

Pseudocode:

```
Auction(b_1, b_2, b_3, ..., b_n):
    Set B = {b_1, b_2, b_3, ..., b_n}
    for i = 1 to n do
        Select b_i ∈ B uniformly at random, without replacement
        if i ≤ ⌈n/2⌉ then
            Reject b_i
        else if i > ⌈n/2⌉ then
            if b_i ≥ max{b_1, b_2, ..., b_⌈n/2⌉} then
                Accept b_i
            end if
        end if
    end for
end
```

Running Time: The for-loop iterates exactly n times and each of the operations within the for-loop require constant time. Therefore, the running time is $O(n)$.

Correctness: Let $b_{max} = \max\{b_1, b_2, \dots, b_{\lceil \frac{n}{2} \rceil}\}$. If there exists a $b_k \in \{b_{\lceil \frac{n}{2} \rceil+1}, b_{\lceil \frac{n}{2} \rceil+2}, \dots, b_n\}$ such that $b_k \geq b^*$, b_k is accepted. The probability of such a b_{max} existing in $\{b_1, b_2, \dots, b_{\lceil \frac{n}{2} \rceil}\}$ **and** a b_k existing in $\{b_{\lceil \frac{n}{2} \rceil+1}, b_{\lceil \frac{n}{2} \rceil+2}, \dots, b_n\}$ is

$$\frac{\left(\frac{n}{2}(\frac{n}{2}-1)!\right)^2}{n!} = \frac{n^2(\frac{n}{2}-1)!(\frac{n}{2}-1)!}{4n!} \geq \frac{n^2(n-1)!(n-1)!}{4n!} = \frac{n(n-1)!}{4} \geq \frac{1}{4} \quad (\text{for } n \geq 2)$$

Thus, The seller accepts the highest of the n bids with a probability at least $1/4$, regardless of n.

Problem 4

Did not have time to complete. Move on to Problem 5.

(a)

(b) i.

ii.

iii. Running Time:

Correctness:

Problem 5

(a) Suppose for a moment that the set S is sorted. Then the position of the median depends on whether the size of the set is odd or even. If $|S| = n$ is odd, the median is located at the mid-point of the sequence of elements in S. The position of the mid-point is the average of the position of the smallest element (i.e., the 1^{st} order statistic) and the position of the largest element (i.e., the n^{th} order statistic). That is, the position of the mid-point lies at $\frac{n+1}{2}$ when $|S| = n$. If $|S| = n$ is even, the sequence of numbers in S has no actual mid-point, so the median is computed by taking the average of the two middle values. In other words, the averages the values at position $\frac{n}{2}$ and the value at position $\frac{n}{2} + 1$. However, S is not sorted, but since the algorithm computes the k^{th} order statistic, we can use it to compute the median.

If $|S| = n$ is odd, we can compute the median by running the algorithm for the $\frac{n+1}{2}$-th smallest value:

$$\text{median}(S) \;=\; \texttt{k-th order statistic}\left(S, \frac{n+1}{2}\right)$$

If $|S| = n$ is even, we can compute the median by calling the algorithm for the $\frac{n}{2}$-th smallest number and again for the $\left(\frac{n}{2}+1\right)$-st smallest number, and then taking the mean of the two values:

$$\text{median}(S) \;=\; \frac{1}{2}\left(\texttt{k-th order statistic}\left(S, \frac{n}{2}\right) \;+\; \texttt{k-th order statistic}\left(S, \frac{n}{2}+1\right)\right)$$

(b) Using the hint, we will find upper bounds for the following:

(i) The expected time of `k-th order statistic` on a subproblem of type j, excluding the time spent on recursive calls, and

(ii) The expected time until an item a_i is selected such that 1/4 of the input can be thrown out, shrinking the input by a factor of 3/4.

Upper bound for (i): For a subproblem of type j, the input size, N_j, is such that $n\left(\frac{3}{4}\right)^{j+1} < N_j < n\left(\frac{3}{4}\right)^{j}$. The for-loop at lines $2-5$ iterates over the entire input, everything else (excluding the recursive calls) requires constant time. Therefore, the expected time spent on the j^{th} subproblem is $O\left(n\left(\frac{3}{4}\right)^{j}\right)$. That is, the expected time spent on the j^{th} subproblem is at most (i.e., bounded above by) $c \cdot n\left(\frac{3}{4}\right)^{j}$, where c is a constant.

Upper bound for (ii): Let X_j be the random variable which denotes the time until a_i is selected such that 1/4 of the input can be thrown out in the j^{th} subproblem. The expected time to find an item a_i such that 1/4 of the input can be thrown out is then

$$\mathbb{E}\left[X_j\right] = \sum_{k=0}^{\infty} k \cdot P\{X_j = k\} = \sum_{k=0}^{\infty} k \cdot (1-p)^{k-1} p$$

where p denotes the probability of finding such an a_i (i.e., the success probability). Therefore, X_j is a geometric random variable. Since X_j is a geometric random variable, the expected time is simply $\frac{1}{p}$. The success probability p and the expected time are both computed below.

If the algorithm throws out 1/4 of the input in the j^{th} subproblem, it must be the case that either 1/4 of the input is smaller than a_i OR 1/4 of the input is greater than a_i.
Let A^- be the event $\{a_i > a_k : a_k \in S^- \text{ and } |S^-| = \left\lceil \frac{|S|}{4} \right\rceil\}$
and A^+ be the event $\{a_i < a_k : a_k \in S^+ \text{ and } |S^+| = \left\lceil \frac{|S|}{4} \right\rceil\}$
Since a_i is uniformly sampled from S at random, the probability of this event is:

$$P\left(A^- \cup A^+\right) \;=\; P\left(A^-\right) + P\left(A^+\right) \;=\; \frac{1}{4} + \frac{1}{4} \;=\; \frac{1}{2}$$

Indeed, 50% of the elements of S satisfy the conditions of such an a_i. Therefore, $p = \frac{1}{2}$.
Since X_j is a geometric random variable, the expected time to find an item a_i such that 1/4 of the input can be thrown out is $\frac{1}{p} = \frac{1}{1/2} = 2$.
Putting it all together, the expected running time for each subproblem (including the recursion) is bounded above by $2cn\left(\frac{3}{4}\right)^j$. Let Y_j be the random variable denoting the running time for the j^{th} subproblem (including the recursive calls). Then $\mathbb{E}\left[Y_j\right] \leq 2cn\left(\frac{3}{4}\right)^j$. Then, letting Y be the *r.v.* that denotes the running time of the entire algorithm and letting M denote the number of subproblems, we have $Y = \sum_{j=1}^{M} Y_j$. Therefore, the expected running time of the entire algorithm is such that

$$\begin{aligned}
\mathbb{E}[Y] = \mathbb{E}\left[\sum_{i=0}^{M} Y_j\right] = \sum_{i=0}^{M} \mathbb{E}\left[Y_j\right] \;&\leq\; \sum_{i=0}^{M} 2cn\left(\frac{3}{4}\right)^j \\
&\leq\; \sum_{i=0}^{\infty} 2cn\left(\frac{3}{4}\right)^j \\
&=\; \frac{2cn}{1-\frac{3}{4}} \\
&=\; 8cn \;=\; O(n)
\end{aligned}$$

Therefore, the expected running time of the entire algorithm is $O(n)$.