Homework 4

Jack Wilburn

November 10th, 2021

1 Q1: QuickSelect

1.1 Using the law of conditional expectation, prove that $T(n) \leq n + \sum_{j=1}^{n} \frac{1}{n} \max T(j-1), T(n-j)$.

The time it takes to complete the algorithm depends on our choice of "pivot". If we choose poorly then we'll have a longer run time and if we choose well, we're done. That said, the expected running time (X_n) depends on the event (e_j) corresponding to choosing a certain pivot we choose. If we don't choose perfectly, we'll need to do n operations to move the elements into the corresponding arrays, B and C. We then have two choices, we look in array B or array C, where the expected run time of those recursive calls are also dependent on e_j. Since we know we're only going to be looking in one of the arrays, the worst case would happen when we get the sub-array that takes the longest to look through. That would be the max of the expected values of the two. To put that into symbols, $\mathbf{E}[X_j|e_j] = n + max(\mathbf{E}[X_{j-1}|e_j], \mathbf{E}[X_{n-j}|e_j])$.

By the law of conditional expectation, and the knowledge that each pivot is chosen with probability $1/n$, we can modify the above equation to $\mathbf{E}[X_j] = n + \sum_{j=1}^{n} max(\frac{1}{n}\mathbf{E}[X_{j-1}], \frac{1}{n}\mathbf{E}[X_{n-j}])$. We can now factor out the $\frac{1}{n}$ to get $\mathbf{E}[X_j] = n + \frac{1}{n}\sum_{j=1}^{n} max(\mathbf{E}[X_{j-1}], \mathbf{E}[X_{n-j}])$. Now, noticing that our sub problems are smaller than the original problem, we can translate the problem into a recurrence relationship: $T(n) \leq n + \frac{1}{n}\sum_{j=1}^{n} max(T(j-1), T(n-j))$

1.2 Using this along with $T(1) = 1$, prove that $T(n) \leq 4n$. Write down a description of all the events you use when you use conditional expectation.

Using $T(1) = 1$ as the base case and assuming that this holds for all values $\geq n$, I'll show that it holds for $n + 1S$. That is, I need to show that $n + \frac{1}{n}\sum_{i=1}^{n} 4i \leq 4n + 4$. We can use the integral trick we saw in class to see take an integral approximation of the summation $n * \sum_{i=1}^{n} \frac{4i}{n} * \frac{1}{n}$, rearranged from above. The integral approximation is $\int_{x=0}^{x=1} 4x dx = 2$. Now we can times that by the n that we placed to the left of the summation when we re-arranged to get $2n$ and we need to remember to add the n from the original equation. Therefore, we have $n + 2n = 3n \leq 4n$. Thus our guessed answer, $T(n) \leq 4n + 4$ is true.

2 Q2: Sampling from a stream

2.1 Prove that in the end, the variable x stores a uniformly random sample from the stream. (In other words, if the stream had N elements, $P(x = a_i] = 1/N$ for all i.)

At $t = 1$, the algorithm has a 100 percent chance of choosing a_1. At time $t = 2$, the algorithm has a 50 percent chance of choosing a_2, and a 50 percent chance of staying with a_1. This continues, with $t = 3$ being slightly more complex where the algorithm chooses a_3 with a $\frac{1}{3}$ probability, but has a $\frac{2}{3}$ probability of staying with the answer from the last step (50 percent a_1 and 50 percent a_2). If we multiply through, we get $x(3) = (1 - \frac{1}{3})(\frac{1}{2}a_1 + \frac{1}{2}a_2) + \frac{1}{3}a_3 = \frac{1}{3}a_1 + \frac{1}{3}a_2 + \frac{1}{3}a_3$ where the coefficients show the chance of each a_x being picked, and x(t) is x at time t. for $t = 4$ it would be $x(4) = (1 - \frac{1}{4})(\frac{1}{3}a_1 + \frac{1}{3}a_2 + \frac{1}{3}a_3) + \frac{1}{4}a_4 = \frac{1}{4}a_1 + \frac{1}{4}a_2 + \frac{1}{4}a_3 + \frac{1}{4}a_4$.

We can generalize this form to following: $x(t) = \sum_{i=2}^{t-1} \prod_{j=2}^{i} (1 - \frac{1}{j})a_i + \frac{1}{t}a_t = \frac{1}{t}a_1 + \frac{1}{t}a_2 ... \frac{1}{t}a_t$ and then plug in n in place of t to show that for any number n, there is a $\frac{1}{n}$ probability of getting that number at the end of the stream.

3 Q3: Walking on a path

3.1 Prove that $T(0) = 1 + T(1)$, and further, that for any $0 < s < n$, $T(s) = 1 + \frac{T(s-1)+T(s+1)}{2}$.

At location v_0, the particle is forced to move to location v_1 taking one step. Thus the solution for the expected time for the particle to move to v_n includes that first step and adds on the the expected time it takes for the particle to get from v_1 to v_n. This is the equation from above, $T(0) = 1 + T(1)$.

At any other location (other than at v_n) v_s, the particle can either move towards v_n or away from it. The particle must move in one direction, so add that step and then if it moved towards v_n, add on the expected number of steps from v_{s+1} for $1 + T(s + 1)$, else add on the expected number of steps from v_{s-1} for $1 + T(s - 1)$. Since both of these outcomes are equally likely, we can multiply both of them by $\frac{1}{2}$ and then add the outcomes together. This yields $T(s) = 1 + \frac{T(s-1)+T(s+1)}{2}$

3.2 Use this to prove that $T(s) = (2s + 1) + T(s + 1)$ for all $0 \leq s < n$, and then find a closed form for $T(0)$.

With a base case of $T(0) = 1 + T(1) = 2(0) + 1 + T(1)$, assume that $T(s) = (2s + 1) + T(s + 1)$ holds for all values $< n$, we'll show that it holds for n.

$T(n) = 1 + \frac{T(n-1)+T(n+1)}{2} = T(s) = 1 + \frac{(T(n)+2(n-1)+1)+T(n+1)}{2}$ by the induction hypothesis.

$\frac{T(n)}{2} = 1 + (n - 1) + \frac{1}{2} + \frac{1}{2}T(n + 1)$ by rearranging and some algebra (subtract $0.5T(n)$ from both sides).

$T(n) = 2(n - 1) + 3 + T(n)$ by multiplying through by 2

$T(n) = (2n + 1) + T(n)$

This is what we we're trying to show.

Using the fact that $T(s) = (2s + 1) + T(s + 1)$, and by plugging in for some smaller values, I guess that the general form for $T(0)$ for a given n is $T(0) = \sum_{i=1}^{n} 2i - 1$. This holds for when $n = 1$ since $(2(0) + 1) + 0 = 2(1) - 1$.

Now assume that for some k, $T(0) = (2(0)+1)+(2(1)+1)+(2(2)+1)+...+(2(k-1)+1)+0 = \sum_{i=1}^{k} 2i-1$, I'll show that it holds for $k + 1$. For $k + 1$, $T(0) = (2(0) + 1) + (2(1) + 1) + (2(2) + 1) + ... + (2(k - 1) + 1) + (2(k) + 1) + 0 \stackrel{?}{=} \sum_{i=1}^{k+1} 2i - 1$. Simplifying things a bit, and using the induction hypothesis, we have $\sum_{i=1}^{k} 2i - 1 + (2(k) + 1) \stackrel{?}{=} \sum_{i=1}^{k} 2i - 1 + (2(k+1) - 1)$. Now I'll subtract the sums from both sides and expand the parentheses to yield $2k + 1 = 2k + 2 - 1$, which is clearly true. Thus by induction, I've proved that the closed form for $T(0) = \sum_{i=1}^{n} 2i - 1$ for any n.

3.3 Give an upper bound for the probability that the particle walks for $> 4n^2$ steps without getting absorbed.

Note that $\sum_{i=1}^{n} 2i - 1 = n^2$. Using Markov's inequality, $P(X \geq 4n^2) = \frac{n^2}{4n^2} = 1/4$ so there is a 25 percent chance that the the particle "walks" $> 4n^2$ steps without getting absorbed.

4 Q4: Birthdays and applications

4.1 What is the expected *number of pairs* (i, j) with $i < j$ such that person i and person j have the same birthday? For what value of n (as a function of m) does this number become 1?

Given m days in a year and a uniform distribution of birthdays, the chance of being born on any particular day is $\frac{1}{m}$. Therefore, the chance of 1 person not sharing a birthday with any other person is $1 - \frac{1}{m}$. The chance that one person doesn't share a birthday with the other $n-1$ people is $(1 - \frac{1}{m})^{n-1}$. The expected number of people with no shared birthday in n people is $\mathbf{E}[X] = n * p = n(1 - \frac{1}{m})^{n-1}$ since it's this probability for all n people at the same time. Therefore, the expected number of people who share a birthday with someone is $n(1 - (1 - \frac{1}{m})^{n-1})$

4.2 Prove that the probability of this happening (conditioned on the library size being a million songs) is < 0.05.

The probability of the radio station playing k distinct songs is $1 * (1 - \frac{1}{1000000}) * (1 - \frac{2}{1000000}) * ... * (1 - \frac{k-1}{1000000}) = \prod_{i=1}^{k} 1 - \frac{(k-1)}{1000000}$. The equation above can be explained in plain English: for the first song there is no song to collide with, then for the second there is one song to collide with, for the third there are two, etc.. For k = 200 the equation returns approximately 0.98. Thus the chance of a collision in this number of songs is approximately 0.02, which is less than 0.05. The fact that there was a collision calls into question, whether there truly are 1 million songs in the library.

5 Q5: Checking matrix multiplication

5.1 Prove that $\Pr[\langle a, x\rangle \neq \langle b, x\rangle (\mathbf{mod} 2)] = 1/2$

Given that A, B are not equal, and a random binary vector, r of the same length as A and B, then $D = Ar - Br$ may equal 0 or 1 (if they were equal it would always be 0, just by factoring). Given that $D = Ar - Br$ may equal 0 or 1, we can say there is some d_i that would equal zero or 1 that would be the determining piece of information that shows whether the algorithm returns true or false.

Since Ar, Br are just vector multiplication, we can expand that out to $p_i = \sum_{k=1}^{n} d_1 + ... + d_i + ... + d_n = d_i + y$. Using the law of conditional expectation we can convert this equation to $P(p_i = 0) = P(p_i = 0 | y = 0) * P(y = 0) + P(p_i = 0 | y \neq 0) * P(y \neq 0)$. We can use that $P(p_i = 0 | y = 0) = P(r_i = 0) = \frac{1}{2}$ and $P(p_i = 0 | y \neq 0) = P(r_i = 1 \wedge d_i = -y) \leq P(r_i = 1) = \frac{1}{2}$ to get $P(p_i = 0) \leq \frac{1}{2} \cdot P(y = 0) + \frac{1}{2} \cdot P(y \neq 0) \ = \frac{1}{2} \cdot P(y = 0) + \frac{1}{2} \cdot (1 - P(y = 0)) \ = \frac{1}{2}$. That's what we were trying to show.

5.2 Now, design an $O(n^2)$ time algorithm that tests if $C = AB$ and has a success probability $\geq 1/2$

Pseudocode

1. Compute a new random vector, d, of length n where each element is 0 or 1 with $p = \frac{1}{2}$
2. Compute $A * (B * d) = k$
3. Check $C * d == k$. If it's true, output true, else false.

Correctness

Let $P(incorrect)$ be the chance that we're incorrect after following this algorithm (i.e. the algorithm shows $AB = C$ when they're not).

First assume that $AB \neq C$, that is for $E = AB - C$, $E \neq 0$. Then there exists some entry of E, call it e_{ij}, such that $e_{ij} \neq 0$. Finally, let $d_{-j} = (d_1, ..., d_{j-1}, d_{j+1}, ..., d_n)$

Given the above assumptions, $P(incorrect) \leq P(E * d = 0) \leq P(\sum_k e_{ik}d_k = 0)$, because the probability that the matrix is zero is less than the probability that the individual multiplication of the rows is zero. $P(\sum_k e_{ik}d_k = 0) = P(e_{ij}d_j = -\sum_k e_{ik}d_k)$ where $k \neq j$. This is equal to $P(d_j = -\frac{1}{e_{ij}} \sum_k e_{ik}d_k)$ where $k \neq j$. Now for the big jump using the law of total probability. The previous equation is equal to $\sum_{x \in 0,1} P(d_j = -\frac{1}{e_{ij}} \sum_k e_{ik}d_k | d_{-j} = x) * P(d_{-j} = x) \leq \sum_{x \in 0,1} \frac{1}{2} * P(d_{-j} = x) = \frac{1}{2}$.

Since $P(incorrect) \leq \frac{1}{2}$ that implies $P(correct) \geq \frac{1}{2}$, which is what we were trying to prove.

Running Time

It takes linear time to generate d, quadratic time to multiply a nxn matrix by a length n vector for $B * d$, $A * (B * d)$, and $C * d$, and linear time again to check $A * (B * d) == C * d$. Thus the largest component here is quadratic, making the overall complexity quadratic.

5.3 Show how to improve the success probability to $7/8$ while still having running time $O(n^2)$

Pseudocode

1. Do the above algorithm above 3 time with 3 different d vectors.
2. Return true if all the 3 trials returned true, else false.

Correctness

The algorithm above is incorrect 50 percent of the time so if we run it 3 times, the chance that it's incorrect lowers to $\frac{1}{2^3} = \frac{1}{8}$, because each trial is independent. Thus, the chance we're correct is $1 - \frac{1}{8} = \frac{7}{8}$

Running Time

The running time here is 3 times the running time above so it's still quadratic.