CS3230: Some useful notes for randomised analysis

Eldon Chung

February 16, 2020

Okay so I mentioned this a bit in my Friday class, so I accidentally left out my Wednesday class on this. I really really want you guys to be able to be able to do one of the take home questions for this coming week but last year when I had to solve it it took the entire hour (and worse still, I tried to do it on the spot. It was a disaster). And in my experience I think part of the reason why it's not so easy is because the question uses tools that really are not that commonly known. Which is a shame, because the result of the proof says something very profound: That perfect hashing is very possible. Now I rather not get into the specifics and turn this document into an essay, but it basically says that if we wanted to store n items in $O(n)$ space, and be able to look them up in $O(1)$ time with easy to compute hash functions, we could totally do it. Perhaps some time during class I can talk about it if time permits. But yes, basically the result of the question is that we can have our cake, and eat it too in $O(1)$ time.

Some Basic Preliminaries

Okay so I hope y'all watched the lecture. If not, please please please please please do. I cannot stress enough. Please do not fall back on the content or else CS3230 will eat you up alive :/ I've already seen it happen to many students. If yoou have, congrats, half of this section should be familiar to you.

Let me start off by talking about random variables. Typically we call them X or Y. Think of these guys sort of like a black box with a button on it. There's a little mystery involved in what they'll give you. You could push the button to find out, and usually these guys will spit out a number. For example, I could define random variables in the following way:

$$X_f = \begin{cases} 1, w.p.\ \frac{1}{2} \\ 0, w.p.\ \frac{1}{2} \end{cases}$$

$$X_u = \begin{cases} 1, w.p.\ \frac{1}{3} \\ 0, w.p.\ \frac{2}{3} \end{cases}$$

But hey, those just kind of look like coins. So you can think of the X's as black box[1] where if you pressed the button, it would reveal its value to you of either 0 or 1. Other examples could include stuff like:

[1] Or pink? I like pink.

$$
\begin{aligned}
Y &= \begin{cases} X+5, w.p. \ \frac{1}{6} \\ X, w.p. \ \frac{5}{6} \end{cases} \\
Z &= \begin{cases} X+5, w.p. \ \frac{1}{6} \\ X, w.p. \ \frac{5}{6} \end{cases} \\
T &= X_u + X_f \\
S &= X_u \cdot X_f \\
U &= X_u^2
\end{aligned}
$$

Now the last two are particularly interesting, since they're random variables defined based on other random variables. We love to do this in CS btw. As if that weren't enough, we also define them recursively sometimes[2].

Now there are two main things I want to mention here. If you've watched the lectures you should undoubtedly be familiar with the *linearity of expectations*, which simply states that, the expectation of a linear combination of random variables is just the linear combination of the expecations. For example:

$$\mathbb{E}[c \cdot X + Y] = c \cdot \mathbb{E}[X] + \mathbb{E}[Y]$$

Oh, and expectations are important for random variables because it's sort of what we think "should happen" if we took the "average" of a lot of experiments. We use this to measure how well we expect our algorithms to perform[3].

Anyway, let's go back to the coins. And in particular, let's talk a little bit about, $X_u + X_f$, $X_u \cdot X_f$, and ${X_u}^2$. Like I mentioned these are really also considered random variables on their own as well. And the way I like to think of these as pressing the buttons of the random variables that they are defined on, taking those values and then evaluating them. So for example, $X_u + X_f$ could evaluate to $0, 1, 2$ depending on what the X_u and X_f evaluate to. ${X_u}^2$ on the other hand, could evaluate to either $0, 1$.

But with what probabilities do these random variables take the values that they do? $X_u + X_f$ takes on the value 1 with probability $\frac{1}{3} + \frac{1}{6}$ (since it must be that exactly one of the two random variables it is defined on evaluates to one when we push their buttons). Now the expected value is, by *LoE*, simply $\mathbb{E}[X_f] + \mathbb{E}[X_u]$.

Now what I want to say about the other two random variables is the main point of this document. It should be a little clear that the something like $X_f \cdot X_u$ by our definitions can only take on either 0 or 1, because the only possible cases of what X_u and X_f evaluate to are $0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0$ or $1 \cdot 1$.

So what's the expected value of those two? You might be tempted to think in general $\mathbb{E}[X \cdot Y] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$, but if you look at $\mathbb{E}[X_u^2]$, you'd notice it's really not the case. I mean if you think about it, $X_u^2 = 1 \iff X_u = 1$, and so $X_u^2 = 1$ with the same probability that X_u is i.e. $\frac{1}{3}$. So $\mathbb{E}[X_u^2] = \mathbb{E}[X_u^2] = \frac{1}{3}$. This does hold for $\mathbb{E}[X_u \cdot X_f]$ though, in particular it indeed is $\mathbb{E}[X_u]\mathbb{E}[X_f]$. So when does it hold? In general if the two random variables are independent, then yes, this holds.

[2] **HENZ INTENSIFIES**

[3] unlike how I let my parent's expectations down during exams

Now the last thing I want to mention is the general paradigm for using indicator random variables (basically random variables that only evaluate to $0, 1$). The reason we use them in CS is because they're seen as "counters". In the sense that you can use them to measure things in a randomised setting. Questions like "how many collisions in the hashtable?" or "how many comparisons does randomised quicksort make?" are all questions that can be handled by simply defining your indicator random variable properly, and then taking a summation.

I'll end this with two examples that demonstrate what I mean, then really hope you guys try the tutorial questions.

Example 1

Let's say I had 3 bins and 2 balls. I'm going to throw both balls into the 3 bins, uniformly at random (this is usually what we call hashing in randomised analysis). What if I wanted to count on average how many balls would bin 1 recieve? Well then I can just define two random variables X_1 and X_2.

$$X_1 = \begin{cases} 1 & \text{if ball 1 falls into bin 1} \\ 0 & else \end{cases}$$

$$X_2 = \begin{cases} 1 & \text{if ball 2 falls into bin 1} \\ 0 & else \end{cases}$$

Then we know that the expected number of balls that falls into bin 1 is just $\mathbb{E}[X_1 + X_2] = \mathbb{E}[X_1] + \mathbb{E}[X_2] = \frac{1}{3} + \frac{1}{3}$. As a small exercise I leave it to you to figure out the reasoning behind each step.

Okay let's try something else. What if I want to count the expected number of collisions, if I had 3 balls and 4 bins? Then one way I can do that is by counting the number of collisions that per pair of items, and then add all them up. Like so:

$$X_{i,t} = \begin{cases} 1 & \text{if ball } i \text{ falls into bin } t \end{cases}$$

Then if I said $Y_{i,j}$ should indicate if item i and j land in the same bin, then that is just the same as $X_{i,1} \cdot X_{j,1} + X_{i,2} \cdot X_{j,2} + X_{i,3} \cdot X_{j,3} + X_{i,4} \cdot X_{j,4}$. You can read this as $Y_{i,j}$ is only 1, when any of the terms $X_{i,t} \cdot X_{j,t}$ are both 1. Again as a small exercise you can try to prove that $Y_{i,j}$ here can only either take on the value 0 or 1[4].

Now we want the expected number of collisions, so that's given as:

$$\mathbb{E}\left[\sum_{i<j}^{3} Y_{i,j}\right] = \sum_{i<j}^{3} \mathbb{E}\left[Y_{i,j}\right]$$

[4] Hint: For example let's say item i and j hashed into bin 3, what should $Y_{i,j}$ evaluate to? What if they both hashed into some other bin? What if they didn't hash into the same bin? Can the sum ever exceed 1? What would that mean if it did?

Now since (I argued that) $Y_{i,j}$ can only take on the values of $0, 1$, then $\mathbb{E}[Y_{i,j}] = Pr[Y_{i,j} = 1]$, which is $4 \cdot \frac{1}{4^2}$. But another way of viewing this is:

$$\mathbb{E}[Y_{i,j}] = \mathbb{E}[X_{i,1} \cdot X_{j,1}] + \mathbb{E}[X_{i,2} \cdot X_{j,2}] + \mathbb{E}[X_{i,3} \cdot X_{j,3}] + \mathbb{E}[X_{i,4} \cdot X_{j,4}]$$

but for example $\mathbb{E}[X_{i,1} \cdot X_{j,1}] = Pr[X_{i,1} \cdot X_{j,1} = 1]$ which is 1 if and only if both $X_{i,1}$ and $X_{j,1}$ are 1, which happens with probability $\frac{1}{4^2}$ (assuming the balls were thrown independently, this is a key assumption I'm making for this analysis to work). So plugging all that back in gives us that computing $\mathbb{E}[Y_{i,j}]$ this way gives us the same result (surprise susprise).

But we're not done. What we wanted was the expected number of collisions. Since we're considering all possible pairs of balls, and there are 3 of them, there's $\binom{3}{2}$ possible pairs $Y_{i,j}$ to consider. But for each it is $4 \cdot \frac{1}{4^2}$. So the expected is $\binom{3}{2} 4 \cdot \frac{1}{4^2}$. Or a little more formally:

$$\begin{aligned}
\mathbb{E}\left[\sum_{i<j}^{3} Y_{i,j}\right] &= \sum_{i<j}^{3} \mathbb{E}\left[Y_{i,j}\right] \\
&= \sum_{i<j}^{3} 4 \cdot \frac{1}{4^2} \\
&= \binom{3}{2} 4 \cdot \frac{1}{4^2}
\end{aligned}$$