1. **Load-balancing**

A given web service has n web servers and one load-balancer that dispatches incoming requests. When a request arrives to the service, the load-balancer randomly (uniformly) chooses which web server to assign the request to and forwards it accordingly.

(a) Consider one particular webserver. What is the probability that no request is assigned to it, given that k requests had come to the load-balancer?

(b) As in the previous part, suppose k requests arrived at the load-balancer. Show that the probability that all servers get assigned at least one request is larger than $1 - ne^{-\frac{k}{n}}$.
You can use the following inequality: $(1-\varepsilon)^x < e^{-x\varepsilon}$ when $x > 0, \varepsilon < 1$.

(c) Now, we want to have a probability p that every server gets assigned at least one request. (If you want, think of the requests as cookies and the servers as EECS70 students in section. We want every student to get at least one cookie.)
Give an upper-bound on the minimal number of requests that need to arrive so that this happens. (How many cookies should a GSI randomly throw out to students so that there is a probability at least p that every single student has at least one cookie?)

(d) Is there a number of requests such that you would be *absolutely sure* that every server is servicing at least one request?

Is your formula in the previous part consistent with this?

(e) What if we wanted to change the desired probability p to make it larger. You could just plug in the new p into the formula.

Instead, try to reason in a simpler direct way to adjust the number of requests required to get to a higher probability $q = 1-(1-p)^{\ell}$ (for some integer ℓ) that every server has at least one request.

Do you think this way of reasoning is conservative for large ℓ?

(f) Suppose we change our goal. Instead of giving every single server at least one request, we want to give every single server at least 10 requests. (Or 10 cookies for each EECS70 student.)

Give an upper bound on the number of requests needed to make that happen with probability at least 0.5.

Feel free to use the following numerical fact: $(1-(1-(1/2))^4)^{10} \geq 0.5$.

(g) Now, you might be thinking that the argument above seems a bit overly conservative in terms of getting each server at least c requests.

So just think intuitively. Suppose I told you that k is the number of requests after which you could reasonably expect to have at least 1 request for in each server.

What would you say is the number of requests required to give each server at least c requests?

Does this differ from your bounds so far? Why?