1 Question 1

Prove that this stupid caching algorithm is k-competitive.

Let ν be a sequence of page requests. Like done so in class, we can divide the request sequence into p phases where a phase stops just before we see $k+1$ distinct pages.

Let's analyze the cost of the stupid algorithm by using the concept of "phases". According to the stupid algorithm, the cache will be completely purged if it is full. Thus, after each phase except the last, the cache becomes full (after ingesting k distinct pages), and will be purged with a cost of k at the start of the next phase (where a new page is requested). The cost for the last phase is 0 since no pages are evicted.

The optimal algorithm, on the other hand, will evict at least 1 page per phase, except for the last one where it might do at least 0 work.

Thus, the ratio of the work done for the first $p-1$ phases by the stupid algorithm A compared to the optimal is $\frac{A(\nu_{p-1})}{OPT(\nu_{p-1})} \leq k$. Suppose that the work done by the stupid algorithm and the optimal algorithm in the last phase is c and d, respectively. Then, we have $A(\nu_{p-1}) + c \leq kOPT(\nu_{p-1}) + kd + c$, which is $A(\nu) \leq kOPT(\nu) + c$. Since $c = 0$ as no pages are evicted by the last phase by the stupid algorithm, we get $A(\nu) \leq kOPT(\nu)$, proving that our stupid algorithm is k-competitive.