Technical University of Crete
School of Electrical and Computer Engineering

Course: **Randomized Algorithms**

Exercise 1

Angelopoulos Dimitris - 2020030038

Coupon Collector In this exercise we will simulate the problem of coupon collector. We assume that we have n different types of coupons to be collected. The selection of some coupon is done uniformly at random. We keep selecting until we collect every coupon.

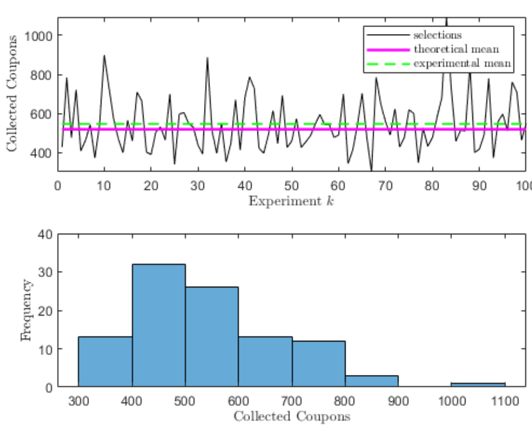
$$X = \sum_{i=0}^{n} X_i$$

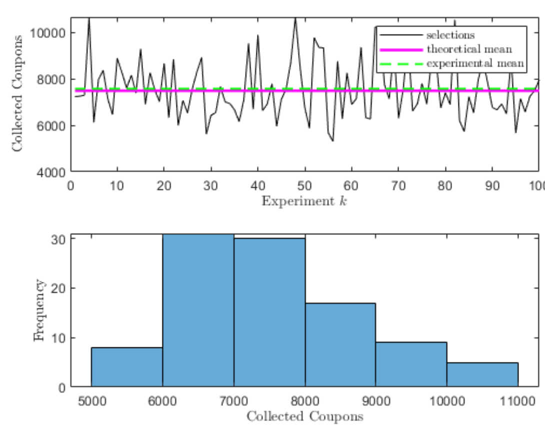
Where X_i is a geometric random variable and represents the number of boxes bought while we had exactly $i-1$ coupons. X is the number of selections we make until we collect every coupon. It turns out that the expected number of selections is given by the following.

$$\mathbb{E}[X] = nH(n) = O(n \log n)$$

Where the harmonic number $H(n) = \sum_{i=1}^{n} 1/i$ satisfies: $H(n) = \ln n + \Theta(1)$.

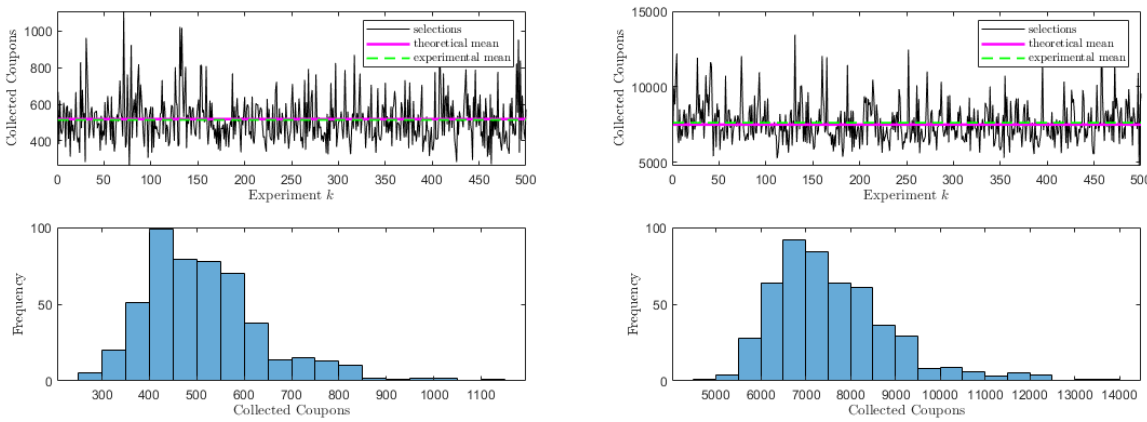
We then simulate the experiment $K = 100, 500$ times. The following results correspond to $n = 100, 1000$ coupons respectively and $K = 100$.





We observe that the experimental expected number of selections is approximately the same as the same with the theoretical one both for $n = 100$(left) and $n = 1000$(right). The histograms for each case seems to be following the binomial distribution.

Know we will solve the problem doing $K = 500$ experiments. That means that the sample size is larger and thus we expect a more accurate result.



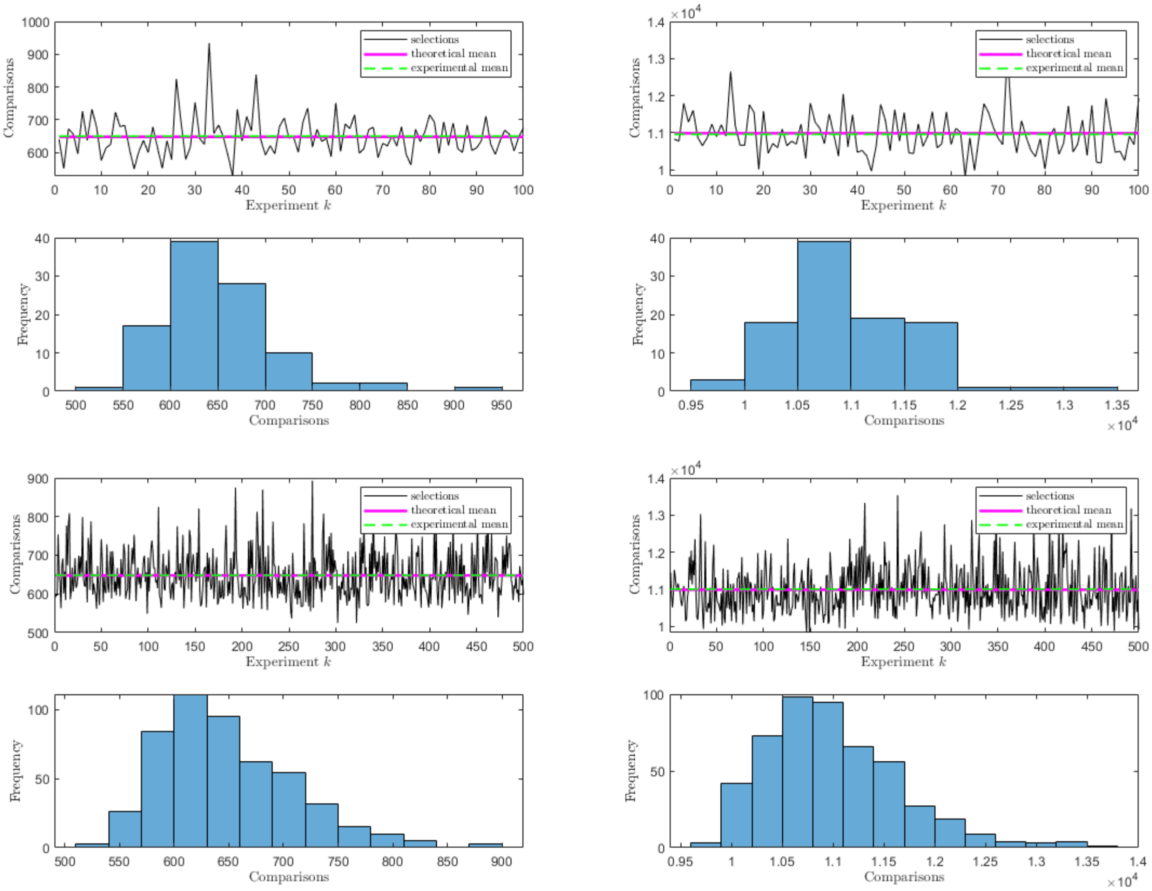
For each n amount of coupons we observe that the experimental mean is (most of the times) the same to the theoretical one. As for the histograms we can verify that they represent the binomial distribution even more accurately (that is because we have access to more samples, as mentioned above).

Quick Sort We will implement the randomized quick sort algorithm. The main difference with its deterministic counterpart is the selection of the pivot element of some distinct list S. The randomization essentially makes the decision of a wrong pivot highly unlikely. The expected number of comparisons X made is the following.

$$\mathbb{E}[X] = 2(n+1)H(n) - 4n = O(n \log n)$$

Following a similar analysis as previously, we make $K = 100, 500$ experiments respectively sorting some random distinct list S of length n.

Much similarly to the coupons collector problem, we check if the experimental mean approximates the theoretical one. The comparisons made by the randomized quick sort algorithm also follows the binomial distribution. For the following plots we have $n = 100$, $K = 100$ (top left), $n = 1000$, $K = 100$ (top right) and $n = 100$, $K = 500$ (bottom left), $n = 1000$, $K = 500$ (bottom right).



Randomized Median In this exercise we will implement the median algorithm using randomization. The expected number of comparisons is given by the following expression.

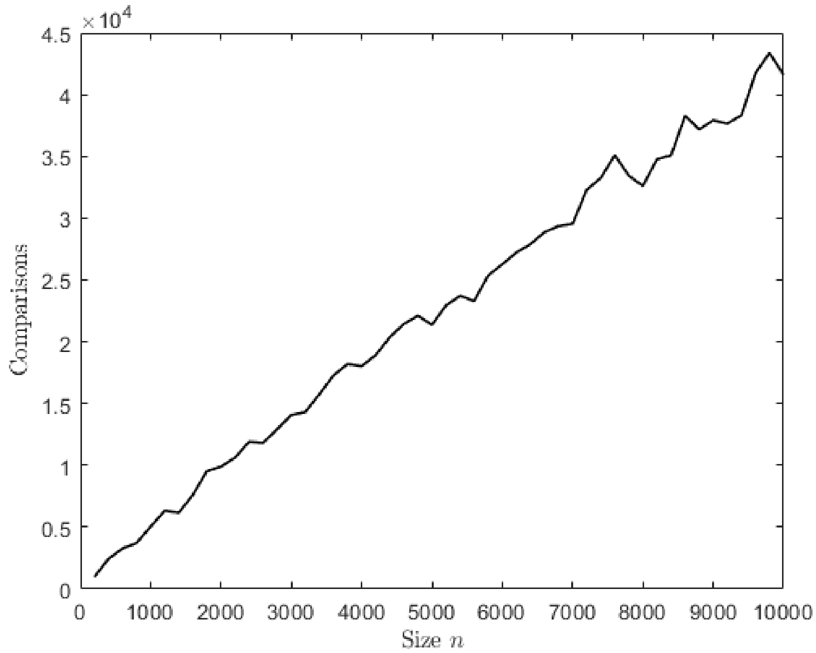
$$\mathbb{E}[X] = T_{sort(R)}(n) + T_{compare(S,d,u)}(n) + T_{check\_FAIL}(n) + T_{sort(C)}(n)$$

We break down how many comparisons are made above. The sorting algorithm used are deterministic and have complexity $O(n \log n)$.

- $T_{sort(R)}(n) = O(|R| \log |R|) = O(n^{3/4} \log n^{3/4}) = o(n)$
- $T_{compare(S,d,u)}(n) = O(n)$
- $T_{check\_FAIL}(n) = O(1)$
- $T_{sort(C)}(n) = O(|C| \log |C|) = o(n)$

Thus we can verify that the randomized median algorithm achieves linear complexity $\mathbb{E}[X] = O(n)$ but with the drawback of having some probability of failure $P(\text{FAIL}) < n^{-1/4}$.

Having said all of the above, we find the median using this algorithm for lists S of length $n \in \{200, 400, \ldots, 10000\}$. If the algorithm fails, for some realization, we repeat until it succeeds.



We observe that the expected comparisons indeed have linear complexity with respect to the length n of the list.