Homework 4

Kevin Chao

November 2019

Question 1

a. Given that P(A) = .5 and $P(A^C)$ = .5, lets make Y the distance traveled, where at minute (n) = 0, Y = 0. Now, when A, when the robot moves north, lets make it Y = Y + 5, and vice versa for A^C, Y = Y - 5. This problem in specific would be considered a binomial, as it could either move north or south, and doesn't stop, and thus the expected movement for each function would be $E[Y_A] = E[Y_{A^C}] = 5np$, where 5 is the movement, n is the number of minutes, and p is the probability. And with the linearity of expectation, $E[X - Y] = E[X] - E[Y]$, we can then do

$$E[Y] = E[Y_A] - E[Y_{A^C}] = 60(.5) - 60(.5) = 0$$

And thus, the E[Y] = 0. The same idea applies for Variance, but instead adds, where

$$Var[Y] = Var[Y_A] + Var[Y_{A^C}]$$

Var[Y_A] can be calculated as np(1-p) = 5*60(.5)(.5), and the same can be said for $Var[Y_{A^C}]$, where np(1-p) = 60(.5)(.5). Thus,

$$Var[Y] = 60(.5)(.5) + 60(.5)(.5) = 150$$

And since the standard deviation is $\sqrt{Var[Y]}$, $\sigma = 12.247$

b. Now, it is more likely to be heading south instead of north. Adjusting for P(A) = .25 and $P(A^C)$ = .75, the following is the process for finding E[Y]:

$$\begin{aligned} E[Y_A] &= 5(60)(.25) \\ &= 75 \\ E[Y_{A^C}] &= 5(60)(.75) \\ &= 225 \\ E[Y] &= 75 - 225 \\ &= -150 \end{aligned}$$

And now lets apply the same thing for Variance to find the standard deviation.

$$\begin{aligned} Var[Y_A] =&5(60)(.25)(.75)\\ =&56.25\\ Var[Y_{A^C}] =&5(60)(.75)(.25)\\ =&56.25\\ Var[Y] =&56.25 + 56.25\\ =&112.5\\ \sigma =&10.607 \end{aligned}$$

c. Assuming that after an hour I am at Y = 0, part(a)'s model would be faster since you are already at the expected position's location and can thus look around the standard deviation area, whereas part(b) requires to walk to the expected position and then look around the standard deviation. However, if you start at the expected location, part(b) would be most sufficient since the standard deviation is lower and would be much faster to search for.

Question 2:

a. Let X be a random variable that shows the given uniform distribution. Since it is a uniform distribution, $E[X] = \frac{0+1}{2} = .5$, and $Var[X] = \frac{(1-0)^2}{12} = .0833$. Plugging into the Central Limit Theorem, the z-score is:

$$Z_n = \sqrt{1000}\frac{.55 - .5}{\sqrt{.0833}} = 5.47$$

With the given z score, the approximate probability of average being greater than .55 is $P(\bar{x} \geq .55) = P(z \geq 5.47 | z \sim N(0,1)) \approx 0.0$. Thus, the random number generator is not correct.

The code to find the probability given the z-score is below:

```
var = 1/12
expected = .5
n = 1000

zscore = n**(1/2)*((.55-expected)/var**(1/2))

print( 1 - stats.norm.cdf(zscore))
```

b. Given that $\bar{x} = .5$, we know that the Z-score is now = 0. Thus, $P(z \geq .5) = .5$ and $P(z \leq .5) = .5$. With the true mean equaling the sample mean and the probability equaling out, that would give me confidence that the RNG is correct.

Question 3:

a. With Markov's inequality,

$$P(X \geq a) \leq \frac{E[X]}{a}$$

However, with a being less than E[X], Markov's makes the upper bound be greater than 1, thus making Markov's not a valid method of determining upper bound in this (since probabilities can never be greater than 1).

b. Markov's inequality gives an upper bound that looks like:

$$P(X \geq a) \leq \frac{4}{a}$$

c. $E[X^2] = \int_0^\infty x^2 f(x)dx \geq \int_a^\infty x^2 f(x)dx \geq \int_a^\infty g(a)f(x)dx = g(a)P(X \geq a)$

$$P(X \geq a) \leq \frac{E[X^2]}{g(a)}$$

We know that $P(X \geq a) \leq \frac{4}{a}$, and thus,

$$\frac{E[X^2]}{g(a)} = \frac{4}{a}$$
$$\frac{1}{g(a)} = \frac{4}{a*E[X^2]}$$
$$g(a) = \frac{a*E[X^2]}{4}$$

d. The equation of variance is known as

$$Var(X) = E[X^2] - E[X]^2$$

and thus can be solved to find $E[X^2]$ given that Var(X) = 10

$$10 = E[X^2] - (4)^2$$
$$26 = E[X^2]$$

Solving for g(a),

$$g(a) = \frac{a*26}{4}$$

Plugging this into part(c)'s inequality,

$$P(X \geq a) \leq \frac{26}{\frac{a*26}{4}}$$

Question 4:

a. Since the sorting algorithm swaps elements in the list around, the worst possible case is if the list starts from largest to smallest, which the number of comparisons would be a sum such that:

$$\sum_{k=1}^{n} k = 1 + 2 + 3 + ... + k = \frac{n(n+1)}{2}$$

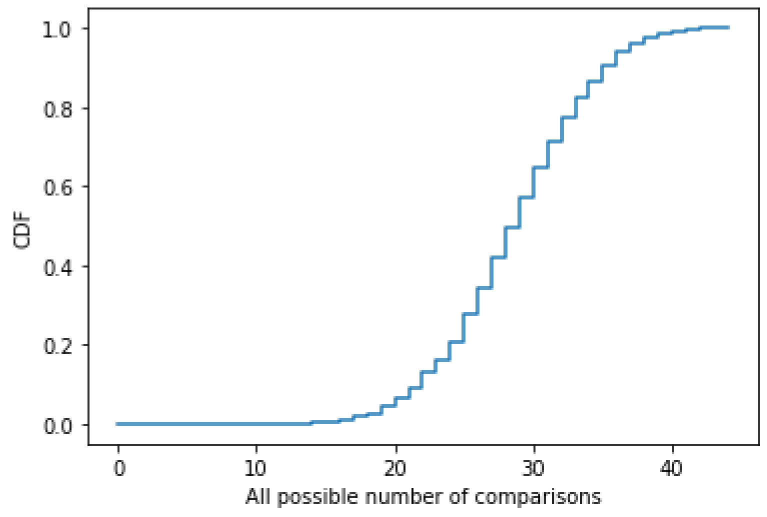
b. Let X_i be the random variable equal to the number of comparisons made by randomized insertion sort at the ith element. X_i would have the probability of being at each position to the left equally at $\frac{1}{i}$. Also at X_i, it could make at most (i-1) comparisons (leftmost/first element) and it has to atleast make 1 comparison (with element next to it), making the range {1,i-1}. With these pieces of information, it can be concluded that:

$$E[X_i] = 1 * \frac{1}{i} + 2 * \frac{1}{i} + ... + (i-1) * \frac{1}{i}$$

Using linearity of expectations, E[X] would then be:

$$E[X] = \sum_{i=1}^{n} E[X_i]$$
$$E[X] = \sum_{i=1}^{n} \sum_{p=1}^{i-1} p * \frac{1}{i}$$

c. The following graphics is the output from my python code of the Monte Carlos Trial given that n = 10.

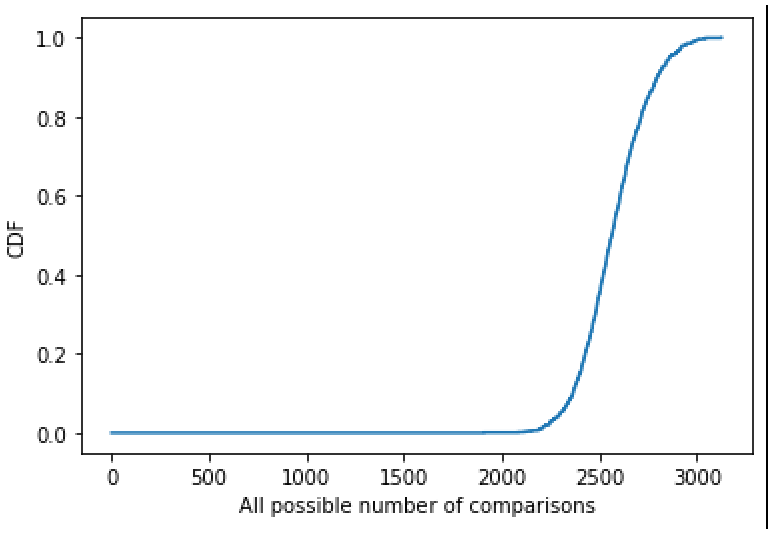


```
Average of Monte Carlo: 29.543
```

The code for the graphics is below.

```
def simulate(n, t=1000):
    return np.array([randomized_insertion_sort(np.random.random(n))[1]
        for _ in range(t)])

def plot_cdf(comparisonList):
    t = len(comparisonList)
    cdf = np.bincount(comparisonList).cumsum()/t
    assert np.max(comparisonList) + 1 == len(cdf)
    assert np.isclose(cdf[-1],1)

    plt.step(np.arange(len(cdf)),cdf)
    plt.xlabel('All possible number of comparisons')
    plt.ylabel('CDF')
    plt.show()

def mean(comparisonList):
    return np.mean(comparisonList)

def run(n, t=1000):
    comparisonList = simulate(n, t)
    plot_cdf(comparisonList)
    print("Average of Monte Carlo: "+ str(mean(comparisonList)))
```

d. The following graphics is the output from the python code of the Monte Carlos Trial given that n=100



```
Average of Monte Carlo: 2571.066
```

As the list length grows, the curve is steeper and tends to develop the curve at higher number of comparisons. If the averages were compared, it shows how the average is leaning towards a higher number $(29.543/40) \approx .75$

and $(2571.066/3000) \approx .8333$