**HW - Week 13**

**Name:** Sahand Sabour **Student ID:** 2020280401

**Page 112:**

5.2-3: Since there are 6 sides to a dice, the probability of getting each side would be

Hence, the expected value for dice i would be

Accordingly, given n dices, since the dices are similar and have the same expected value, we have that

**Page 142:**

5.4-2: Initially, given that all the bins are empty, we need to toss at least two balls and at most b+1 balls (pigeonhole principal) to have a bin contain two balls. Accordingly, for the kth toss we know that k-1 balls have been tossed to empty bins since the condition hasn’t been met yet. The probability of tossing the ball into one of the k-1 non-empty bins at the kth toss would be while the probability of the ball being tossed into an empty been would be calculated via the following multiplication, since after a bin is occupied the numerator decreases by 1 until all bins have been filled, which gives the overall probability as

Accordingly, the expected number of tosses would be

As mentioned previously, considering that the probability of having a bin with two balls after b+1 tosses is 1 due to the pigeonhole principal, the above equation can be rewritten as

**Page 144:**

**5-2.h:** In the case that k=0, then the condition A[i]=x is never met, which makes the worst running time n. The expected running time would also be n since all of the elements have to be scanned to ensure that the no element satisfies the condition.

If k=1, the worst running time occurs when x is at the last position of the array, which takes n since all the elements have to be scanned. For the expected running time, since x could be in either of the n positions of A, the probability of x being in a position i would be

Hence, let m denote the position of X in the array; the expected value of scans to find X would be

**5-2.i:** I would use DETERMINSTIC-SEARCH. The average time for RANDOMIZED\_SEARCH would be n(ln(n)+O(1)) whereas the the average expected running time for the DETERMINSTIC\_SEARCH and SCRAMBLE\_SEARCH is n, as previously shown. Moreover, generating the permutations in SCRAMBLE\_SEARCH has a linear time complexity n, which makes DETERMINSTIC\_SEARCH more efficient.

**Page 204:**

8.4-4: The area of the circle is . We can consider the circle as a set of n rings, each having an area of . Let us assume that each ring starts at distance of the center and ends at radius . Hence, it can be derived that

which can be considered as a recurrence relation that gives

Accordingly, the general case would be

Each point could be assigned to a ring if

Hence, we would have n buckets and each point would be assigned to a bucket if it satisfies the above condition for a ring.