Question 1

Prove that the sum of the n first odd positive integers is n^2, i.e., 1 + 3 + 5 + · · · + (2n − 1) = n^2

Answer:

Let S(n) = 1 + 3 + 5 + · · · + (2n − 1). We want to prove by induction that for every positive integer n, S(n) = n^2.

1. Basis Step: If n = 1 we have S(1) = 1 = 1^2, so the property is true for 1.
2. Inductive Step: Assume (Induction Hypothesis) that the property is true for some positive integer n, i.e.: S(n) = n^2.

We must prove that it is also true for n+1, i.e., S(n + 1) = $(n + 1)^2$. In fact:
S(n + 1) = 1 + 3 + 5 + · · · + (2n + 1) = S(n) + 2n + 1 .
But by induction hypothesis, S(n) = n^2, hence:
S(n + 1) = n^2 + 2n + 1 = $(n + 1)^2$.
This completes the induction, and shows that the property is true for all positive integers.

Let A and B be two events. Suppose that the probability that neither A or B occurs is 2/3. What is the probability that one or both occur?

Answer:

If P((A ∪ B)^C) = 2/3, the complementary event is P(A ∪ B) which would then be 1/3
because P((A ∪ B)^C) + P(A ∪ B) = 1

Question 2

Let A be the following event:
If given two dice, what is the probability that the sum of the two numbers rolled will be at least 9?
Answer:
Each pair has the probability of 1/36. There are 10 pairs whose sum is at least 9.
Solution: 10/36

	1	2	3	4	5	6
1	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
2	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
3	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	**(3,6)**
4	(4,1)	(4,2)	(4,3)	(4,4)	**(4,5)**	**(4,6)**
5	(5,1)	(5,2)	(5,3)	**(5,4)**	**(5,5)**	**(5,6)**
6	(6,1)	(6,2)	**(6,3)**	**(6,4)**	**(6,5)**	**(6,6)**

Let B be the following event:
If given two dice, what is the probability that the two numbers rolled will be the same?

Answer:
Each pair has the probability of 1/36. There are 6 pairs with the same number.
6/36

	1	2	3	4	5	6
1	**(1,1)**	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
2	(2,1)	**(2,2)**	(2,3)	(2,4)	(2,5)	(2,6)
3	(3,1)	(3,2)	**(3,3)**	(3,4)	(3,5)	(3,6)
4	(4,1)	(4,2)	(4,3)	**(4,4)**	(4,5)	(4,6)
5	(5,1)	(5,2)	(5,3)	(5,4)	**(5,5)**	(5,6)
6	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	**(6,6)**

What is the probability of B given A?

P(B|A) = $\frac{P(A \cap B)}{P(A)}$

Answer:

The event of A intersecting B occurs twice, for pairs (5,5) and (6,6). Each has a probability of 1/36. Thus $P(A \cap B)$ = 2/36.

The probability of A is 10/36.

Solution:

P(B|A) = $\frac{P(A \cap B)}{P(A)} = \frac{\frac{2}{36}}{\frac{10}{36}} = \frac{2*36}{10*36} = \frac{1}{5}$

Question 3: Counting

There are 6 chipmunks and 7 zebras in a ballroom dancing class. If 4 chipmunks and 4 zebras are chosen and paired off, how many pairings are possible? (Note: each zebra and chipmunk is considered a unique, separate being).

$\binom{6}{4} * \binom{7}{4} * 4!$

We choose 4 chipmunks from the 6. Then we choose the 4 Zebras. And then we pair them off. Note that we multiply the values together.

Question 4: Pointers

Determine the output of the following code:

```
int f(int* n, int m){
       *n = 10;
       m = 10;
       return *n + m;
}



int main(){
       int n = 5;
       int m = 5;
       int res = f(&n, m);
       cout << res + n + m << endl;
```

```
}
```

```
Answer: 35
```

Question 5: Counting

There are 100 balls in a bucket:

- 30 red balls numbered 1, 2, 3, …, 30.
- 70 green balls numbered 1, 2, 3, …, 70.

In how many ways can you pick 20 balls, such that there will be exactly *k* red balls (without replacement)?
Explain your answer.

- Let A be the event of getting exactly *k* red balls. To find |A| we need to find out how many ways we can choose *k* red balls and 20-k green balls.
- There are a total of 30 red balls so to get *k* red balls, there are $\binom{30}{k}$ ways and order does not matter.
- The remaining 20-*k* selections all must be green balls in order to meet the requirement of "exactly *k* red balls".
- Use the product rule to combine because the sets are mutually exclusive.
- Therefore there are

$\binom{30}{k} * \binom{70}{20-k}$ ways to pick 20 balls such that there will be exactly *k* red balls.

Question 6: Expectation

In the following game, a fair coin is tossed until either a head comes up or four tails come up. Let X be the random variable that denotes the number of tosses made in the game.

a. Find the distribution of X. That is, for each possible value of X, say what is the probability X would get that value.
b. What is E[X]? That is, find the expected value of X.

Explain your answers.

Possible outcomes of the game: (4 cases)

X=1 H
X=2 TH
X=3 TTH

X=4 TTTH
X=4 TTTT

p(X=1) = 1/2
p(X=2) = 1/2 * 1/2 = 1/4
p(X=3) = 1/2 * 1/2 * 1/2 = 1/8
p(X=4) = 1 - 1/2 - 1/4 - 1/8 = 1/8

E[X] = 1*1/2 + 2*1/4 + 3*1/8 + 4*1/8 = 1.875

Question 7: Algorithm Analysis

a)

```
void f(int* n, int* m, int n_size, int m_size) {
    for (int i = 0; i < n_size; i++) {
        for (int j = 0; j < m_size; j++) {
          // Some O(1) operation here
          }
     }
}
```

$\Theta(nm)$

b)

```
void f(int n){
   for (int i = 0; i < n; i++) {
       for (int j = 0; j < i; j++) {
        //O(1) work here
       }
   }
}
```

$\Theta(n^2)$

c)

```
void f(int n){
   for (int i = n; i >= 0; i /= 2) {
       for (int j = 0; j < 1000; j++) {
           //O(1) operations
       }
```

```
    }
}
```
$\Theta(\log n)$

Question 8: Coding

Given an array of numbers, write a function to move all 0s to the end of the array while maintaining the relative order of the non-zero elements. Do this **in-place**. This should run in $\Theta(n)$.
Example: [0,2,0,1,0] -> [2, 1, 0, 0, 0]

```
void moveZeroes(int nums[], int n) {
      int j = 0;
     for (int i = 0; i < n; i++) {
          if (nums[i] != 0) {
             nums[j++] = nums[i];
          }

}

     for (int i = j; i < n; i++) {
          nums[i] = 0;
} }
```

Given an array of nums, find the length of the longest sequence of zeroes **recursively**. You can use the `std::max` function.
Example: `maxZeroLength([0,0,1,0,0,0], 6, 0) = 3`

```
int maxZeroLength(int nums[], int len, int startIdx){
   //base case -->
   if(startIdx == len){
       return 0;
   }
   int maxLen = 0;
   while (startIdx < len && nums[startIdx++] == 0){
       maxLen++;
   }
   return max(maxLen, maxZeroLength(nums, len, startIdx));
}
```