***1.10.* Найти случайное событие X из равенства Х + А + Х + А = В**

Х + А + Х + А = В

2X + 2A = B

X + A = B/2

X = B/2 – A

***3.11.*** **На отрезке длиной l независимо одна от другой поставлены две точки, положение каждой из которых равновозможно на этом отрезке. Определить вероятность того, что из трех получившихся частей одного отрезка можно построить треугольник.**

Для того, чтобы из трех отрезков можно было построить треугольник, необходимо выполнение неравенства треугольника: сумма длин любых двух отрезков должна быть больше длины третьего отрезка.

Рассмотрим отрезок длиной l и две независимо поставленные на нем точки A и B. Будем считать, что A левее B.

Обозначим длины полученных отрезков как a = |AB|, b = |O1A|, c = |O2B|, где O1 - левый конец отрезка, O2 - правый.

Изображение выглядит как зарисовка, линия, диаграмма, белый

Автоматически созданное описание

Тогда неравенство треугольника запишется как:

a + b > c

a + c > b

b + c > a

Из первого неравенства следует, что точка B не может лежать правее середины отрезка l, т.е. c <= l/2.

Из второго неравенства следует, что точка A не может лежать левее середины отрезка l, т.е. b >= l/2.

Из третьего неравенства следует, что a < l/2, что выполняется автоматически, так как a = c - b, c <= l/2, b >= l/2.

Таким образом, вероятность того, что из трех получившихся частей отрезка можно построить треугольник, равна вероятности того, что точка B лежит правее середины отрезка, а точка A левее середины отрезка. Эта вероятность равна произведению вероятностей каждого из описанных событий, т.е. 0.5 \* 0.5.

Ответ: вероятность равна ¼ = 0.25.

Проверим результат, смоделировав данную ситуацию при помощи программы на языке Python.

from random import randint  
  
len = 10000000000000 # количество дискретных точек на прямой  
tr = 100000000 # количество тестов  
pos = 0 # счётчик положительных результатов  
  
for i in range(tr):  
 if (i / tr \* 100 % 1 == 0): print(i / tr \* 100)  
 p1 = randint(0, len+1) # определим точки при помощи RNG  
 p2 = randint(0, len+1)  
 l1 = min(p1, p2) # длина первого отрезка равна координате левой из двух точек  
 l2 = max(p1, p2) - min(p1, p2) # длина среднего отрезка равна разности координат точек  
 l3 = len - max(p1, p2) # длина последнего отрезка равна разности координаты конца отрезка и коор. правой точки  
  
 if max(l1, l2, l3) < l1 + l2 + l3 - max(l1, l2, l3): # для определения положительного результата имеет смысл сравнивать лишь наибольший отрезок с суммой остальных  
 pos += 1   
print("-------------------------------------")  
print(pos/tr)

Результат работы программы:



Очевидно, результат моделирования совпадает с теоретическим