**Лабораторная работа №1-2**

**Анализ алгоритмов и их сложности**

**Цель работы:** провести анализ алгоритма и оценить его сложность. Приобретение навыков исследования временной сложности алгоритмов и определения ее асимптотических оценок.

Вариант 16

в формуле вынесем x-ы за скобки и перепишем многочлен в виде

Pn(x) = a0 + x(a1 + x(a2 + ... ( ai + .. x(an-1 + anx))).

Расстановка скобок диктует такой порядок вычислений:

S0 = an,

S1 = S0 x + an-1,

S2 = S1 x + an-2,

…,

Si = Si-1 x + an-i,

…,

Sn = Sn-1 x + a0, Pn(x) = Sn.

Рассмотренный метод называется схемой Горнера. На рис. 5 приведена блок-схема алгоритма вычисления значения многочлена по схеме Горнера.



Рисунок 5 - Вычисления значения многочлена по схеме Горнера (алгоритм 2)

Подсчет временной сложности в этом случае гораздо проще: для вычисления Si требуется 1 умножение и 1 сложение. Всего такая итерация осуществляется n раз. Таким образом, временная сложность этого алгоритма: Т(п) = n умножений + n сложений = 2n операций.

То есть таким образом мы видим то что алгоритм линейный (О(n)) и время выполнения такого алгоритма 1 секунда

Задание 2

Это pyton

def calc\_polinom(A,n,x):

S = A[n]

i = 1

while i <= n:

S = S \* x + A[n-i]

i += 1

return S

supermassive\_black\_hole = [1, 2, 5, 1, 8, 9, 4]

print(calc\_polinom(supermassive\_black\_hole, 6, 10))