Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки
Алгоритми та структури даних
Лабораторна робота №1.2
“Алгоритми з вкладеними циклами та метод динамічного програмування”

Виконав:
студент групи ІВ-71
Мазан Я. В.
Залікова книжка №ІВ-7109
Перевірив Сергієнко Анатолій Михайлович

Київ, 2017р.

Тема: алгоритми із вкладеними циклами та метод динамічного програмування

Мета: засвоєння теоретичного матеріалу та набуття практичних навичок використання різних циклічних керуючих конструкцій, вкладених циклів, методу динамічного програмування та обчислення кількості операцій алгоритмів.

Постановка задачі:

1. Задане натуральне число n. Вирахувати значення заданої
2. формули за варіантом.
3. Для вирішення задачі написати дві програми:
    1. Перша програма повинна використовувати для обчислення формули вкладені цикли;
    2. Друга програма повинна виконати обчислення формули за допомогою одного циклу з використанням методу динамічного програмування.
4. Виконати розрахунок кількості операцій для кожного з алгоритмів за методикою, викладеною на лекції, додавши до неї підрахунок кількості викликів стандартних функцій.
5. Програма має правильно вирішувати поставлену задачу при будь-якому заданому n, для якого результат обчислення може бути коректно представлений типом double.
6. Результуючі дані вивести у форматі з сімома знаками після крапки.

Завдання мого варіанту:

9. $$S = \sum_{i=1}^{n} \frac{(2^i + 1)^2}{\prod_{j=1}^{i} (j + 1)}$$

Хід роботи:

Текст програми 1 (вкладені цикли):

```
#include <stdio.h>
#include <math.h>

int main() {
	int i,j,n;
	double S,cyclej;
	printf("Input n: ");
	scanf("%i", &n);
	S = 0;
	for (i=1; i<=n; ++i) {
```

```
            cyclej=1;
            for (j=1; j<=i; ++j) {
                cyclej *= j+1;
                }
            S += pow((pow(2,i)+1),2)/cyclej;
        }
        printf("%.7f\n",S);
}
```

Текст програми 2 (динамічне програмування):

```
#include <stdio.h>
#include <math.h>

int main() {
        int n,i;
        double S,production;
        S=0;
        production=1;
        printf("\nInput n: ");
        scanf("%i", &n);
        for (i=1;i<=n;++i) {
            production *= i+1;
            S += pow(pow(2,i)+1,2)/production;
            }
        printf("%.7f\n",S);
}
```

Тестування програм:

	Програма 1	Програма 2
n=1	4.5000000	4.5000000
n=2	8.6666667	8.6666667
n=3	12.0416667	12.0416667

Еталон: n=3

$$S = \frac{\left(2^1+1\right)^2}{1+1} + \frac{\left(2^2+1\right)^2}{(1+1)(2+1)} + \frac{\left(2^3+1\right)^2}{(1+1)(2+1)(3+1)} = \frac{9}{2} + \frac{25}{6} + \frac{81}{24} = 12.041(6)$$

Обрахунок складності програми 1:

	i =	1	2	3		n
Операції	Додавання	1+1+1	2+1+1	3+1+1		n+2
	Множення	1	2	3		n
	Ділення	1	1	1	• • •	1
	Піднесення до степеня	2	2	2		2

$$T(i) = \sum_{i=1}^{n} (n+2) + \sum_{i=1}^{n} n + n \cdot 1 + n \cdot 2 = \frac{3(n+2)}{2} n + \frac{1 \cdot n}{2} n + 3n =$$

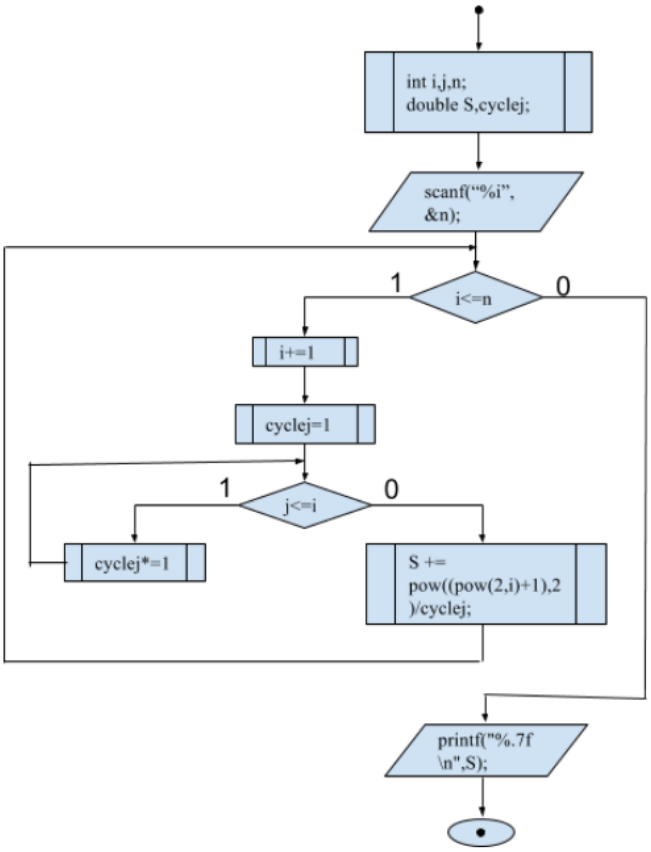
$$= \frac{3n^2}{2} + 6n + 3n + \frac{n^2}{2} = 2\,n^2 + 9n = O\left(n^2\right)$$

Обрахунок складності програми 2:

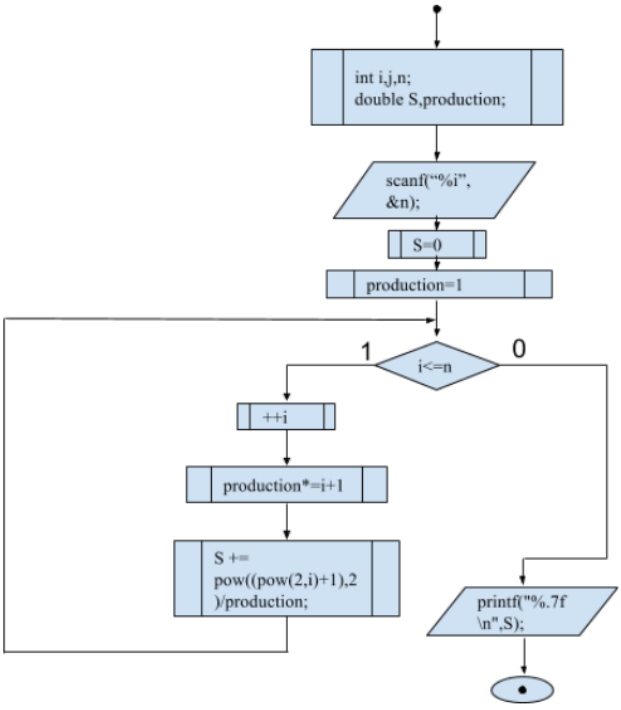
	i =	1	2	3		n
Операції	Додавання	3	3	3		3
	Множення	1	1	1		1
	Ділення	1	1	1	• • •	1
	Піднесення до степеня	2	2	2		2

$$T\,(i) = 3 \cdot n + 1 \cdot n + 1 \cdot n + 2 \cdot n = 7n = O\,(n)$$

Блок-схема програми 1:



Блок-схема програми 2:



Результати розрахунків обома програмами для заданого значення n.

n	Програма 1	Програма 2
-5	0.0000000	0.0000000
0	0.0000000	0.0000000
2	8.6666667	8.6666667
17	17.5068747	17.5068747
100	17.5068754	17.5068754
10000	-nan	-nan

Висновок:

Під час виконання даної лабораторної роботи я навчився використовувати метод динамічного програмування як один із видів оптимізації алгоритму та обчислювати їхню складність. Виконання даної лабораторної роботи допомогло мені краще зрозуміти та освоїти вивчений матеріал.