

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра технічної кібернетики

ДИСКРЕТНІ СТРУКТУРИ  
ЛАБОРАТОРНА РОБОТА №1

ТЕМА: АТД

Виконали:  
Студенти гр. ІТ-82  
Бірюченко О.  
Жук О.  
Михайлова Д.  
Кавун Ф.

Перевірив:  
Сергєєв Д. С..

Київ – 2019

## 1. Обраний АД:

Стек та Однозв'язний лінійний список

## 2. Складність алгоритму та псевдокод:

```
... ... @@ -0,0 +1,32 @@
1 + Псевдокод для стеку:
2 +
3 + (4) Add(x, stack):
4 +   (1) temp := new Stack
5 +   (1) temp->element = x;
6 +   (1) temp->next = stack->start;
7 +   (1) stack->start = temp;
8 +
9 + (1) file fin;
10 + (n) while !fin.endOfFile():
11 +   (1) x := fin.read()
12 +   (4) Add(x, stack)
13 + -----
14 + max(1 + n * (1 + 4)) = max(1 + 5n) = O(n)
15 + -----
16 +
17 + Псевдокод для списку:
18 +
19 + (4) Add(x):
20 +   (1) temp := new element;
21 +   (1) temp->x = x;
22 +   (1) temp->Next = Head;
23 +   (1) Head = temp;
24 +
25 + (1) file fin;
26 + (n) while !fin.endOfFile():
27 +   (1) x := fin.read()
28 +   (4) list.Add(x)
29 + -----
30 + max(1 + n * (1 + 4)) = max(1 + 5n) = O(n)
31 + -----
32 +
```

### 3. Програма:

```
#include "pch.h"
#include <iostream>
#include <fstream>
#include <string>
#include <ctime>

using namespace std;

struct Stack
{
    int element;
    Stack *next, *start;
};

void Add(int x, Stack *&Stack_link)
{
    Stack *temp = new Stack;
    temp->element = x;
    temp->next = Stack_link->start;
    Stack_link->start = temp;
}

// used for debug
/*void Show(Stack *Stack_link)
{
    Stack *temp = Stack_link->start;

    while (temp != NULL)
    {
        cout << temp->element << " ";
        temp = temp->next;
    }
}*/

struct element
{
    int x;
    element *Next;
};

class List
{
    element *Head;
public:
    List() { Head = NULL; }
    void Add(int x);
    void Show();
};

void List::Add(int x)
{
    element *temp = new element;
    temp->x = x;
    temp->Next = Head;
    Head = temp;
}

// used for debug
/*void List::Show()
{
    element *temp = Head;
```

```

        while (temp != NULL)
        {
            cout << temp->x << " ";
            temp = temp->Next;
        }
    }*/

int main()
{
    cout << "-----Stack algorithms time measurement-----\n";
    cout << "Adding elements to stack time:\n";

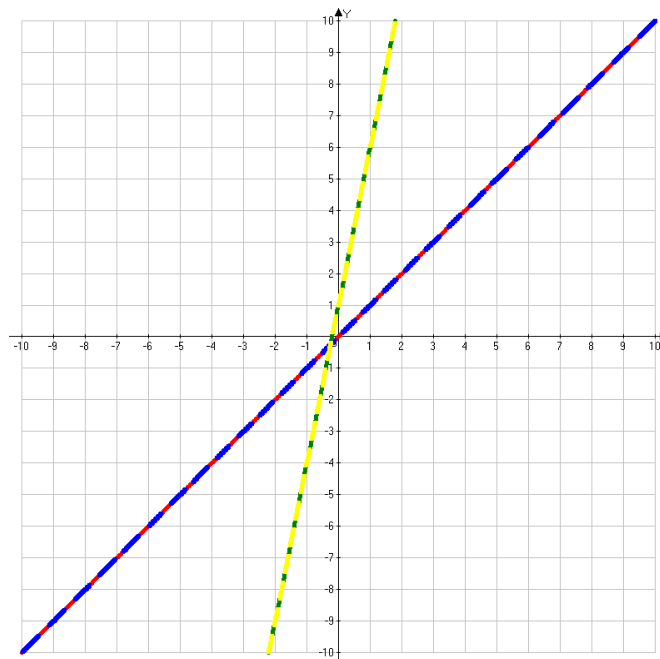
    Stack *Stack_link = new Stack;
    Stack_link->start = NULL;

    ifstream fin;
    int n = 10;
    unsigned int scores[10];
    unsigned int sum = 0;
    int x;
    unsigned int start_time, finish_time;
    for (int i = 0; i < n; i++) {
        fin.open("input.txt");
        start_time = clock();
        while (!fin.eof()) {
            fin >> x;
            Add(x, Stack_link);
        }
        finish_time = clock();
        fin.close();
        scores[i] = finish_time - start_time;
        sum += scores[i];
        cout << "loop " << i << ": " << scores[i] << " ms\n";
    }
    cout << "Average time: " << sum / n << " ms\n";
    cout << "-----\n";

    cout << "-----Single linked list time measurement-----\n";
    List list;
    sum = 0;
    for (int i = 0; i < n; i++) {
        fin.open("input.txt");
        start_time = clock();
        while (!fin.eof()) {
            fin >> x;
            list.Add(x);
        }
        finish_time = clock();
        fin.close();
        scores[i] = finish_time - start_time;
        sum += scores[i];
        cout << "loop " << i << ": " << scores[i] << " ms\n";
    }
    cout << "Average time: " << sum / n << " ms\n";
    cout << "-----";
}

```

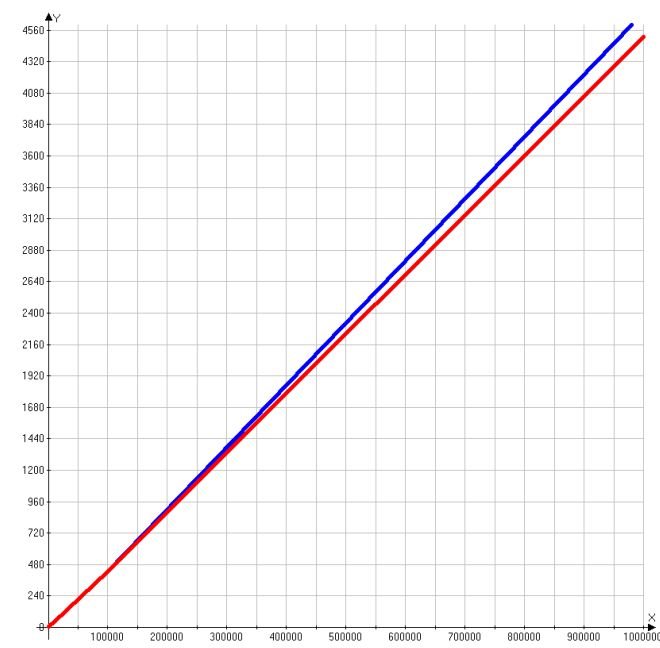
### 3. Графіки та час виконання



Графік 1,2

Червоний - складність стеку  $O(n)$ ; синій - складність списку  $O(n)$ ;  
жовтий - формула складності стеку; зелений - формула складності списку

Час виконання:



Графік 3

Синій - час виконання для стеку; червоний - час виконання для списку

```
Консоль отладки Microsoft Visual Studio
-----Stack algorithms time measurement-----
Adding elements to stack time:
loop 0: 418 ms
loop 1: 409 ms
loop 2: 431 ms
loop 3: 406 ms
loop 4: 411 ms
loop 5: 401 ms
loop 6: 408 ms
loop 7: 405 ms
loop 8: 406 ms
loop 9: 404 ms
Average time: 409 ms
-----
-----Single linked list time measurement-----
loop 0: 402 ms
loop 1: 403 ms
loop 2: 407 ms
loop 3: 405 ms
loop 4: 403 ms
loop 5: 401 ms
loop 6: 403 ms
loop 7: 408 ms
loop 8: 401 ms
loop 9: 404 ms
Average time: 403 ms
-----
D:\Study\Discrete Structures\lab1\Debug\lab1.exe (процесс 10356) завершает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу...
```

Час додавання елементів у стек та додавання елементів у однозв'язний список приблизно однаковий, що і показало визначення складності алгоритмів, а також підтвердилося роботою програми.

Для кожного алгоритму було зроблено по 10 тестів з кількістю елементів 100 000 для отримання більш точного часу виконання.