# Розрахункова робота

**ПОРІВНЯЛЬНИЙ АНАЛІЗ ПРОДУКТИВНОСТІ РОБОТИ СТРУКТУР ДАНИХ ТА ДОСЛІДЖЕННЯ АЛГОРИТМУ СОРТУВАННЯ**

**Мета роботи**  виконати порівняльний аналіз продуктивності роботи структур даних та дослідження алгоритму сортування

# Порядок виконання роботи

1. Виконати порівняльний аналіз продуктивності роботи лінійного списку (лаб. роб. 1) і системного неузагальненого та узагальненого стеку/ черги в залежності від варіанту. Для цього:
   1. Доопрацювати свою лаб. роб. 1 таким чином, щоб була можливість додавати данні у вигляді структури/ об’єкту типу ключ- значення.
   2. Заміряти час додавання та видалення n-даних (n>10000).
   3. Заміряти час додавання та видалення n-даних (n>10000) для системного неузагальненого стеку/ черги.
   4. Заміряти час додавання та видалення n-даних (n>10000) для системного узагальненого стеку/ черги.
   5. Побудувати графіки залежності часу додавання/ видалення даних від кількості елементів для реалізації лаб. роб. 1, неузагальненого та узагальненого стеку/ черги. На одній координатній площині додавання для 3-х структур, на другій видалення. Кількість точок для побудови графіка 5-10.
2. Доопрацювати свою лаб. роб. 6 таким чином, щоб була можливість задавати данні у вигляді структури/ об’єкту типу ключ- значення і відсортовувати в залежності від значення ключа. Заміряти час виконання в залежності від числа елементів і початкового виду масиву. Розглянути ситуації:
   1. Елементи в масиві розташовані у випадковому порядку
   2. Елементи в масиві вже відсортовані
   3. Елементи в масиві розташовані у зворотному порядку
   4. Побудувати графіки з даними досліджень.
   5. Побудувати графіки з даними досліджень (з графіка на одній координатній площині). Для побудови графіків можна використовувати Excel.

# Зміст звіту

Зміст звіту з розрахункової робоит має включати наступні підрозділи:

1. Титульний аркуш.
2. Зміст (генерується автоматично).
3. Постановка завдання (із зазначенням номеру свого варіанту, структурою, що досліджується та назвою алгоритму сортування).
4. Звіт повинен мати:
   1. Опис класів необхідних для заміру часу виконання операцій.
   2. Проектування необхідних класів і структур (допустимо для кожної задачі окремо).
   3. Екранні форми роботи програми (допустимо для кожної задачі окремо).
   4. Графіки побудовані за результатами досліджень tср=f(n).
   5. Висновки за результатами досліджень.
   6. Машинний лістинг програми

# Приклад заміру часу

//Створення змінної watch

System.Diagnostics.Stopwatch watch = new System.Diagnostics.Stopwatch(); watch.Start();//запуск лічильника

//початок фрагменту програми   
int j = 0;

for (int i = 0; i < 10000000; i++)

j++;

//кінець роботи 1-го фрагменту програми

//виведення на екран лічильника

label1.Text = watch.ElapsedTicks.ToString(); for (int i = 0; i < 10000000; i++)

j++;

watch.Stop();//зупинка лічильника

//виведення загального лічильника label2.Text = watch.ElapsedTicks.ToString();