

# Звіт до лабораторної роботи №2

## Виконав Грицина Антон, К-27

Було досліджено операцію `adjacent_difference` з політиками виконання `seq`, `par`, `unseq`, `par_unseq` та як час виконання залежить від них.

Було досліджено вплив кількості потоків виконання при паралельній обробці власного алгоритму на час виконання.

Алгоритм розбиває вибірку на  $K$  рівних частин та додаткову частину з залишком даних, які не можна було розподілити порівну для заданого розміру послідовності.

Оскільки мій процесор має 12 потоків, було проведено дослідження для  $K$  від 1 до 36, на векторах даних розміром 1, 10 та 100 мільйонів.

Також код було протестовано за оптимізацій `/O0`, `/O1`, `/O2`.

За результатів коду зроблено такі висновки:

1. Найшвидшим бібліотечним алгоритмом зазвичай є `par`, а найповільнішим стандартний `seq`.
2. Час виконання для власного алгоритму завжди повільно +- зменшується, наближаючись до  $K$  кратного 12, а після цього різко росте.
3. Зазвичай найбільш оптимальне  $K$  трохи менше або дорівнює числу, кратному кількості потоків процесора, потоки починають заповнювати виділені для них потоки процесора, а час, який витрачається на перемикання між ними, в моменті відвертає приріст швидкості.

У текстових файлах `outputO0.txt`, `outputO1.txt` та `outputO2.txt` наведено вивід коду за різних оптимізацій.