

Advanced Algorithms Programming Assignment #1

Implement and Measure the Speed of Insertion sort, Selection sort, and Merge sort

Due date: **2022.10.24 11:59 PM KST**

- 런어스를 통해 제출

TA: 신유진(raki2001@yonsei.ac.kr)

본 과제에서는 대표적인 정렬 방법인 *Insertion sort*, *Selection sort*, 그리고 *Merge sort*를 구현하고 이들의 수행 시간을 측정해 본다. 이때, 총 세가지 자료형에 대해 정렬을 지원해야 하는데, 각각 정수, 문자(문자열 X), **Student** 구조체 자료형이다. 이를 위해, 제시된 세 가지 자료형 별로 **compare** 함수를 작성하고, 이를 **sort** 함수에 파라미터로 전달하여 대소비교를 할 수 있도록 한다. 정수는 정수의 크기, 문자는 아스키코드에 따라서 대소비교를 진행하며, **Student** 구조체는 **name**이 아닌 **id**로 정렬할 수 있도록 한다. 'in' 폴더에서 각 자료형의 입력 파일 예시를 확인할 수 있다.

Instruction

- 런어스에 업로드된 압축파일을 다운받아 진행.
- **make** 커맨드를 통해 프로그래밍 한 내용을 컴파일하는 걸 권장.
- **./sort** 커맨드를 통해 실행하여 스스로 입력값을 넣는 테스트를 해볼 수 있음.
- **make test** 커맨드를 통해 채점 내용 자가 채점 권장 (**주의: 실제 채점시에는 다른 입력 파일이 적용됨**).
- 스켈레톤 코드는 자유롭게 수정가능.

Requirements

- Implementation MUST be in C
- Your implementations MUST work in Ubuntu. (WSL 과 VM ware 모두 사용가능)

Report (in PDF format)

- 아래와 같은 내용들을 포함.
 - What you did
 - 과제 내용에 대한 요약, 고찰, 그리고 실행시간에 대한 그래프
 - What you learned
 - 과제 수행 과정에서 어떤 것을 배웠는지
 - 어떤 부분이 쉽고 어떤 부분이 어려웠는지

Grading

제출은 **리포트와 .c 파일만** 학번.zip으로 압축할 것. (**비주얼 스튜디오 솔루션 파일 X**)

- Insertion sort (30 points)
- Selection sort (30 points)
- Merge sort (30 points)
- Report (10 points)

*각 자료형에 대해 10 점씩 부여함. 예컨대 **insertion sort**가 숫자, 문자열에서는 잘 동작하지만 구조체에서 동작하지 않는 경우 20 점만 부여

[Optional] Extra credit task

- Java, Python 3, 혹은 Rust 를 이용해 위와 같은 작업을 수행할 경우 추가 점수 부여 (**최대 50 pt**)