

어셈블리프로그래밍설계및실습 – Term Project

임의의 부동 소수점으로 이루어진 배열을 병합 정렬(merge sort)과 삽입 정렬(insertion sort) 두 가지로 정렬하고 메모리에 저장하는 코드를 구현해 서로 비교한다.

1. Introduction

임의의 숫자를 생성해 저장하는 코드가 주어진다. 해당 코드를 이용해 10000 개의 숫자를 생성해 저장하고, 이를 정규화 된 부동 소수점 배열로 보고, 병합 정렬과 삽입 정렬을 구현한다.

2. Algorithm

- 삽입 정렬

<p>Insertion Sort Execution Example</p>	<pre> INSERTION-SORT (A) for j ← 2 to length[A] do key ← A[j] Insert A[j] into the sorted sequence A[1 . . j - 1]. i ← j - 1 while i > 0 and A[i] > key do A[i + 1] ← A[i] i ← i - 1 A[i + 1] ← key </pre>
삽입 정렬 예시	삽입 정렬 수도코드

- 병합 정렬

	<pre> MERGE-SORT(A, p, r) 1 if $p < r$ 2 $q = \lfloor (p + r)/2 \rfloor$ 3 MERGE-SORT(A, p, q) 4 MERGE-SORT($A, q + 1, r$) 5 MERGE(A, p, q, r) MERGE(A, p, q, r) 1 $n_1 = q - p + 1$ 2 $n_2 = r - q$ 3 let $L[1..n_1 + 1]$ and $R[1..n_2 + 1]$ be new arrays 4 for $i = 1$ to n_1 5 $L[i] = A[p + i - 1]$ 6 for $j = 1$ to n_2 7 $R[j] = A[q + j]$ 8 $L[n_1 + 1] = \infty$ 9 $R[n_2 + 1] = \infty$ 10 $i = 1$ 11 $j = 1$ 12 for $k = p$ to r 13 if $L[i] \leq R[j]$ 14 $A[k] = L[i]$ 15 $i = i + 1$ 16 else $A[k] = R[j]$ 17 $j = j + 1$ </pre>
병합 정렬 예시	병합 정렬 수도코드

3. Specification

- 프로젝트 구현 조건

- ✓ 난수 생성시 무한대, NaN 등의 예외 구현
- ✓ 정렬하고자 하는 임의의 배열의 길이에 관계없이 작동하여야 한다.
- ✓ 주어진 코드의 Do not change this area 부분은 바꾸지 말아야 한다.
- ✓ 메모리를 추가로 사용하는 것은 관계없으나, 최종 결과물은 라벨 final_result_series 부터 시작한다.
- ✓ 프로그램은 주석을 포함해 다음과 같이 종료한다

◆ MOV pc, #0 ;Program end

4. Notify

- 최종 결과 : 12 월 6 일 수요일 23:59 까지 klas 과제 제출. 아래 파일들을 zip 으로 압축하여 제출
- 제출 양식 : Assembly_Project_학번_이름.zip

(예시 : Assembly_Project_2023000000_홍길동.zip)

- ✓ 보고서 : Assembly_Project_학번_이름.pdf
- ✓ Code : Term_project.s
- ✓ memory.ini
- 베이스 코드 :
<https://drive.google.com/drive/folders/1sl8WCZXMORBfnAFZmPWcoQBFyQT6B7Q1?usp=sharing>

5. Report Outline

- 발표 : Powerpoint 로 발표 희망자에 한해 10 분 분량으로 작성
- 보고서 : 최소한 다음의 내용들이 포함되어야 하며 그 외의 것을 추가하는 것은 자유

1. Introduction

- 프로젝트에 대한 간단한 설명 작성

2. Background

- 과제를 해결하기 위한 선행 지식들에 대한 설명 (삽입, 병합 정렬)

3. Algorithm

- 프로젝트의 문제를 해결하기 위한 접근 법과 해결한 방식에 대한 설명

4. Performance & Result

- 프로젝트의 완성된 결과 사진과 이에 대한 설명
- States 는 임의의 수를 저장하는 부분 이후부터 계산한다.

5. Consideration

- 과제 중 발생한 문제점과 이를 해결한 방법 등을 서술

6. Reference

6. Grade

- 코드: 35%
- Performance: 35%
- 결과보고서: 30%