올림차순(ascending)과 내림차순(descending) 코딩 과제

정보대학 - 컴퓨터학과

2024100027

최민성(CUI MINCHENG)

1. 알고리즘 개요(Algorithm Overview)

   본 프로그램은 버블 정렬(Bubble Sort) 알고리즘을 포인터를 사용하여 구현하였습니다. while(1) 루프는 정렬이 완료될 때까지 전체 Pass 를 반복하며, 내부의 for 루프는 인접한 두 요소를 비교합니다. is_zero 변수는 최적화 플래그(Optimization Flag)로 사용되어, 한 번의 패스 동안 데이터 교환(Swap)이 발생하지 않으면 정렬이 완료된 것으로 판단하고 무한 루프를 종료합니다.

2. 포인터의 활용 (Usage of Pointers)

   배열 Index (ary[i])를 사용하는 대신, 포인터 변수 ptr 을 선언하여 메모리 주소에 직접 접근하였습니다.

   ptr = data; 가 배열의 시작 주소를 포인터에 할당했습니다. *ptr 가 현재 가리키는 주소의 값을 참조합니다.

   *(ptr + 1) 는 현재 주소의 바로 다음 int 형 데이터 값을 참조합니다.

   ptr++ 는 포인터 주소를 증가시켜 다음 요소로 이동합니다. 이를 통해 배열의 모든 요소를 순차적으로 Find 합니다.

3. 단일 함수 제약 조건 해결(Handling the Single Function Constraint)

   과제의 핵심인 "함수는 1 개만 사용 가능"을 준수하기 위해, sort 함수에 mode 라는 변수를 추가하였습니다. 별도의 오름차순/내림차순 함수를 만들지 않고, 하나의 함수 내에서 if (mode == 1) 조건문과 else if (mode == 2) 조건문을 사용하여 비고 연산자(> 또는 <)를 동적으로 선택하도록 작성하였습니다.

   올림차순은 *ptr > *(ptr + 1) 일 때 교환합니다.

   내림차순은 *ptr < *(ptr + 1) 일 때 교환합니다.

4. 값의 교환 (Swap Logic)

   조건이 충족될 때 tmp 변수를 사용하여 두 Memory Address 의 값을 Swap 합니다. tmp = *ptr; , *ptr = *(ptr + 1); , *(ptr + 1) = tmp; 과정을 통해 데이터의 위치를 Swap 합니다.