**알고리즘 과제 1**

소프트웨어학부 20223908학번 김주영

IDE: CLion, 언어: C99, 컴파일러: GCC(권장), MSVC

**배열 생성 함수에 대한 분석**

배열을 생성하는 makeArray()함수에 대한 분석이다.

배열에 idx변수 값을 기준으로 좌우에 정렬된 데이터를 할당한다. 이는 동일한 두 개의 반복문을 통해 구현 되어있다. 또한 반복문의 내부 구현을 보면 j의 값을 통해 배열에 값을 할당한다.

idx는 rand()함수로 값이 정의되며 가질 수 있는 값의 범위는 0~n-1이다. 이는 idx가 0일 때, 오직 하나의 정렬된 배열로 정의될 수 있다는 것을 의미한다.

또한 두 반복문의 내부를 보면 j를 기존 값에 1~10의 난수를 더하여 정의한다. 이는 반환할 배열의 연속한 두 값이 1~10의 차이를 가진다는 것을 의미한다.

첫 번째 반복문이 끝나고 두 번째 반복문을 실행할 때 j를 0으로 초기화하기 때문에, 이 때에는 증가하는 수열이 아니게 된다. (idx가 0인 경우 제외)

**주어진 배열에 대한 최적의 정렬 알고리즘**

정렬 알고리즘을 크게 비교와 대입 연산으로 나눌 수 있다. 본 알고리즘의 주요 사항은 불필요한 비교연산을 줄이는 것이다.

특정한 기준 점(=배열 생성기의 idx변수 값)을 기준으로 나눈 양쪽의 두 부배열은 이미 정렬이 된 상태이다. 따라서 이미 알고 있는 사실(각 부배열은 이미 정렬이 되어있다)을 비교연산을 통해 또 확인할 필요가 없다.

이를 감안하여 고안한 알고리즘은 다음과 같다.

1. 정렬된 배열을 나누는 기준점을 찾는다.
2. 찾은 기준점을 기준으로 두 배열을 나눈다.
3. 각 배열에서 가장 작은(가장 왼쪽 인덱스에 있는) 원소끼리 비교한다.
4. 더 작은 값을 결과 배열에 할당한다.
5. 과정 (4)에서 사용한 값을 포함하는 배열에서 해당 값을 제거한다.
6. 두 부배열 중에서 남은 원소가 있다면 과정 (3)으로 돌아간다.

**알고리즘 구현**

위 알고리즘의 동작 과정 중에서 인간 언어에 더욱 가까운 부분에 한해서 구현을 소개하고자 한다.

(2). 개념적으로 나눈다는 것이며 실질적으로는 정렬된 두 배열 각각에 대해서 시작위치를 저장하는 변수를 선언한 것뿐이다.

(5). (2)에서 설명한 바와 같이 인덱스 관리를 통해 해결한다. 사용한 배열의 현재 인덱스를 저장하는 변수의 값을 1 증가시키면 된다. 이로써 해당 변수에 저장하고 있는 값보다 작은 인덱스는 탐색 대상에서 제거하겠다는 의미이다.

**알고리즘 평가**

최선의 경우(배열 생성할 때 idx가 0이라 하나의 정렬된 배열만 생긴 경우) 반복 횟수는 N(과정(1) ), 최악의 경우(idx가 최대 범위인 n-1인 경우) 반복 횟수는 2N(과정(1), 과정(3)~(6) )이다.

이를 big-Oh notation으로 표기하면 최악, 최선의 경우 모두 O(N)이다.

**결과**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

위 결과는 Windows PowerShell에서 컴파일, 실행한 결과이다.

My Sort함수 실행 결과 정렬이 되었고 데이터 결과값의 합이 일치하는 것을 확인할 수 있다. 성능 또한 Insertion Sort, Selection Sort보다 빠른 것을 확인할 수 있다.

cmd에서 같은 명령어를 통해 컴파일할 수 있고, Developer Command Prompt for VS에서

“cl main main.c”

명령어를 통해 MSVC로 컴파일하여 main.exe 실행파일을 생성할 수 있다.