

2021 Spring OOP Assignment Report

과제 번호 : Programming Assignment #1_problem 2

학번 : 20200437

이름 : 김채현

Povis ID : kch3481

명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

문제 2번

1. 프로그램 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- "input.txt" 파일에서 오름차순으로 정렬할지, 내림차순으로 정렬할지를 정하는 문자열을 입력받고, 다음 줄에서 정렬할 원소들을 입력받는다.
- 원소들을 배열에 저장하고 Merge sort algorithm을 이용하여 배열 안 원소들을 오름차순으로 정렬한다.
- 입력받은 정렬방법이 UP이면 오름차순, 즉 배열 안 원소를 순서대로 출력파일에 출력하고, DOWN이면 내림차순, 즉 배열 안 원소를 맨 끝 값부터 반대순서로 출력파일에 출력한다. 위의 과정을 입력파일에서 입력받을 값이 없을 때까지 반복한다.

1) data file

data file은 정렬할 방법과 원소들이 나열되어 있는 입력파일인 "input.txt"가 필요하다. 또한 프로그램 실행 후에는 정렬된 값이 적혀있는 출력파일 "output.txt"가 생성된다. 입력파일을 읽어들이는 과정으로는 ifstream을 이용하였다. 반대로 출력파일을 읽어들이는 과정으로는 파일포인터를 이용하였다.

2) 입력부/처리부/출력부

입력부 : 정렬방법과 정렬할 원소를 입력받는다.

처리부 : string으로 읽어들이는 정렬할 원소를 배열 number에 삽입, 배열을 분할, 크기대로 합병정렬, string으로 읽어들이는 정렬방법이 UP인지 DOWN인지, 입력받을 값이 더 이상 없는지를 판단, 처리해야 한다.

출력부 : 정렬된 원소들을 오름차순인지 내림차순인지에 따라 순서대로 출력한다.

2. 프로그램 구조 및 알고리즘

1) Merge sort

Merge sort(합병 정렬)은 다음 단계들로 이루어진다. 먼저, 분할 단계로 입력 배열을 같은 크기의 2개의 부분 배열로 분할한다. 다음은 정복 단계로 부분 배열을 정렬한다. 이 때 부분 배열의 크기가 충분히 작지 않으면(single part가 아니면) 순환 호출을 이용하여 다시 분할 정복 방법을 적용한다. 마지막은 결합 단계로 정렬된 부분 배열들을 하나의 배열에 합병한다. 이 때가 실제로 정렬이 이루어지는 시점이다. 위 알고리즘을 예시를 통해 그림으로 나타내면 다음과 같다.

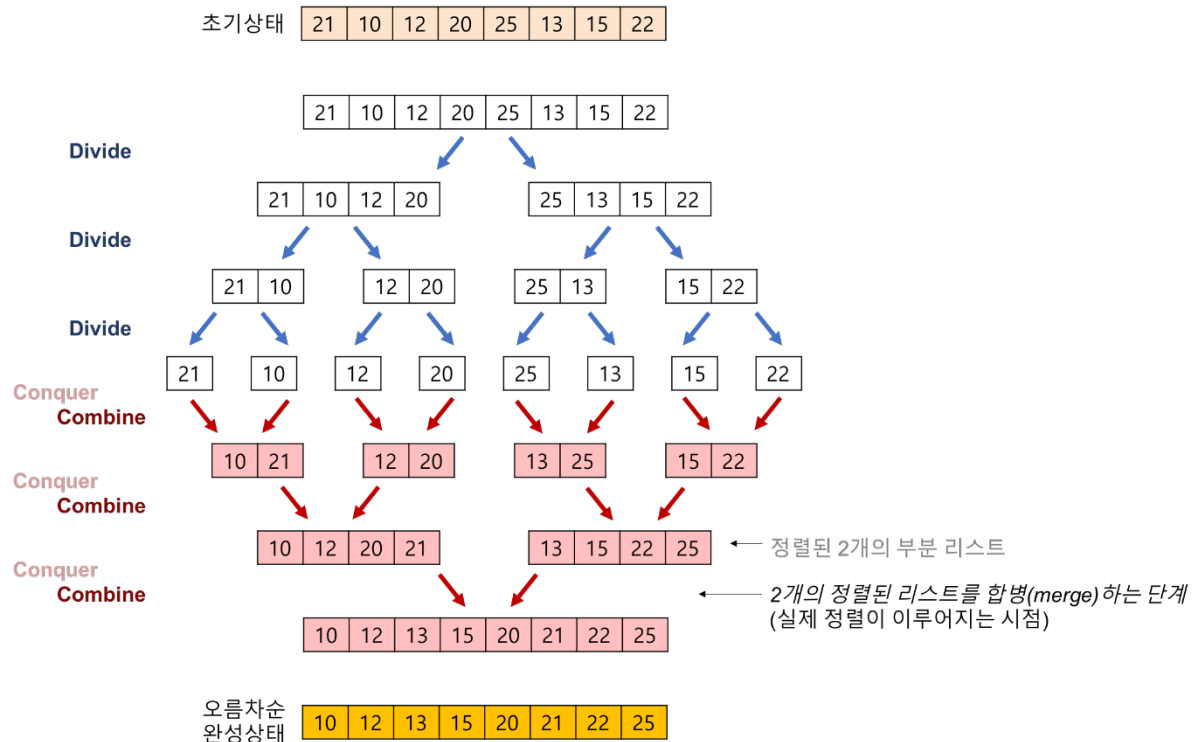


그림 1. Merge sort algorithm

2) 변수 `**unsigned`

(1) `ifstream infile`

입력파일을 받기 위해 `ifstream` 클래스로 선언한다. (`ifstream`은 프로그램 파일에 있는 어떠한 것을 스트림 버퍼에 가지고 와서 프로그램에 입력해주는 클래스이다.)

(2) `string sort_type, data`

입력파일에서 읽어온 정렬방법은 `sort_type`에, 그 다음 줄에 있는 정렬 원소들은 `data`에 저장한다. `getline` 함수를 이용하여 한 줄씩 읽어와 `string`으로 각각 저장한다.

(3) `stringstream line`

`string` 문자열에서 값을 읽어오기 위해 `stringstream` 클래스로 선언한다. 하나씩 읽어온 값들은 벡터에 차례로 저장된다.

(4) `vector<int> number`

정렬할 원소값들을 문자열로 저장한 `string data`에서 `stringstream line`을 이용하여 `int temp`에

임시로 저장하고 이를 push_back 함수를 이용하여 벡터 number에 저장한다.

(5) unsigned int data_size

정렬할 원소의 개수를 저장하는 변수로서, temp 값을 벡터 number에 저장하는 반복문이 돌아가는 동안 임시 변수 i에 대해 i++을 해주고 반복문을 나오면 i값을 data_size에 저장한다. Vector의 사이즈를 구하는 방법인 number.size()를 사용해도 되지만 vector 관련 함수를 최소한으로 사용하기 위해 사용하지 않았다. 0이상의 수만 저장하기 때문에 최대한 많은 정렬 원소를 받기 위해 unsigned int 타입으로 선언하였다.

(6) unsigned int left, right, middle

Merge sort에서 분할 혹은 병합하고자 하는 vector의 부분 중 시작지점, 즉 젤 왼쪽의 원소자리를 뜻하는 변수 left와 끝지점, 즉 젤 오른쪽 원소자리를 뜻하는 변수 right이다. middle은 left와 right의 중간지점을 의미하며, 쉽게 (left+right)/2라고 해주어도 되지만 overflow를 막기 위해 left+(right-left)/2로 계산해주었다. 0이상의 수만 저장하기 때문에 최대한 많은 정렬 원소를 받기 위해 unsigned int 타입으로 선언하였다.

(7) vector <int> sort_number

Merge sort에서 병합부분을 담당하는 함수 combine에서 선언되는 벡터로서, 크기 순서를 비교하여 작은 순서대로 임시로 원소를 저장하는 용도로 사용된다. 병합과정이 끝나면 sort_number에 있는 원소들 순서대로 number에 다시 저장해준다. **

(8) FILE* outfile

파일포인터로 출력파일을 열기 위해 outfile을 선언하였다.

3) 함수

(1) division

Merge sort에서 분할(divide) 부분을 담당한다. division 함수 내에서도 분할된 부분 배열이 하나의 원소로 충분히 작지 않으면 recursion으로 계속하여 분할해준다. 이를 위해 division 함수는 left<right일 때만 실행될 수 있도록 조건문을 걸어주어 left==right가 될 때까지 분할해준다. 배열의 중간 부분인 middle에 대해서 다시 반으로 나누고, 계속 반으로 나누는 과정을 통해 완벽히 분할한다. 분할된 부분배열은 크기가 최소 2인 배열부터 combine함수를 호출하여 정렬, 합병한다.

(2) combine

분할된 부분 배열 두 개를 합병하는 단계이다. 이 단계에서 실제로 정렬이 일어난다. 이 때 정렬한 원소를 임시로 저장하기 위한 배열 sort_number가 필요하다. 합병 정렬 방법은 다음과 같다. 2개의 부분 배열의 값들을 처음부터 하나씩 비교하여 두 개의 배열 값 중에서 더 작은 값을 sort_number로 옮긴다. 두 부분배열 중 하나가 끝날 때까지 이 과정을 반복한다. 둘 중 하나의 부분배열이 먼저 끝나면 나머지 배열의 값들을 전부 sort_number로 복사한다. sort_number을 원래 배열 number로 옮긴다.

4) 이 외 구현 세부 사항

(1) vector

처음에는 정렬할 원소의 개수를 모르기 때문에 크기가 INT_MAX인 배열을 선언하여 사용하려 했지만 배열의 크기가 너무 커 디버깅이 되지 않는 문제가 발생하였다. 이를 해결하기 위해 자동으로 메모리가 할당되는 배열이라고 할 수 있는 vector을 사용하였다. Vector은 container 용도로만 사용하였으며, input.txt 파일에서 받아온 원소를 저장하는 벡터 number과 combine 함수에서 정렬된 원소를 임시로 저장할 벡터 sort_number를 선언할 때 사용해주었다.

3) 토론 및 개선

Merge sort algorithm을 구현하는데 포인트가 되는 것은 순환 호출, 즉 재귀함수(recursion)을 제대로 사용할 줄 아는가이다. "프로그래밍과 문제해결(CSED101)"에서 재귀함수에 대해 배우지 않았기에 처음에 구현하는 데에 헛갈림과 어려움이 있었지만 이번 계기를 통해 recursion에 대한 개념을 확실히 정리할 수 있어서 도움이 되었다. 또한 본 문제에서 중요한 점은 얼마나 많은 수의 원소를 정렬할 수 있냐라고 생각하는데, 처음 배열을 정의하여 사용했지만 정렬할 수 있는 데이터 개수가 한계가 있다는 문제가 발생하였다. 본 코드도 데이터 개수를 받는 변수 data_size를 unsigned int로 선언하여 한계는 있지만 충분히 많은 데이터를 받을 수 있을 것이라 생각된다. 자료형의 최대 크기와 값에 대해 생각해보게 된 좋은 기회였다. 아쉬운 점은 입력파일에서 정렬할 원소를 string에 저장 후 vector에 저장하는 방법이 아닌 바로 vector에 저장한다면 프로그램 실행 속도가 빨라질 것으로 생각된다.

4) 참고 문헌

<https://gmlwjd9405.github.io/2018/05/08/algorithm-merge-sort.html> - [알고리즘] 합병 정렬(merge sort)이란 (그림 1 출처)