3주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20191619 이름: 이동석

**1. 실습 RangeArray**

알고리즘을 설명하기에 앞서 Array.h를 알아야한다. 이 헤더파일에는 Array라는 class가 정의되어있다. class Array는 다음 protected 와 public으로 나뉜다. protected는 배열 포인터인 int \*data와 배열의 크기인 int len으로 이루어져있다. public은 다음과 같다.

Array(int size) : 배열크기를 전달받는다.

1. size<0 이라면, 에러를 출력

2. 그렇지 않다면, size크기를 가진 배열을 생성

~Array() : 프로그램이 끝나기 전 소멸자로 메모리를 해제

int length() const : 배열의 크기를 반환

int& operator[](int i) : [] 연산자 재정의

1. 인덱스가 범위 내에 있다면 값을 리턴

2. 그렇지 않다면, 에러를 출력하고 tmp를 리턴

int operator[](int i) const : [] 연산자 재정의

1. 인덱스가 범위 내에 있다면 값을 리턴

2. 그렇지 않다면, 에러를 출력하고 0을리턴

void pirnt() : 배열의 원소 출력

RangArray의 알고리즘은 다음과 같다.

1. RangeArray 는 Array를 상속한다.

2. RangeArray은 배열의 시작과 끝을 입력 받는다.

3. 배열의 크기는 (끝 – 시작 + 1) 이며 이를 Array(int size)로 배열을 생성한다.

4. baseValue()는 low (시작)을 리턴하고, endValue()는 high(끝)을 리턴한다.

5. RangeArray의 i번째 값은 Array 배열에서 (i-low)값과 동일하다. 따라서, i-row 값을 반환받아야한다. 이를, Array에서 operator로 정의한 []를 이용한다.

위 프로그램의 공간 복잡도는 RangeArray의 size와 같다. 따라서 공간복잡도는 high-row+1 이다. 시간복잡도는 Big-‘oh’로 생각해보면, 단순하게 배열에 값을 넣는 동작이므로, 최소 O(high-row+1)로 생각할 수 있다.

**2. 과제 Str**

Str.h 헤더파일에는 class Str이 정의되어 있다. class Str은 private과 public으로 이루어져있다. private은 string의 내용을 담고있는 char \*str과 string 길이를 저장하고 있는 int len으로 이루어진다. public은 다음과 같다.

Str(int leng) : string의 길이 leng을 전달 받는다.

1. leng < 0 이라면 에러를 출력한다.

2. 그렇지 않다면, 크기가 leng+1 str 배열을 생성하고, 첫 문자에 문장의끝을 의미하는 ‘\0’을 할당한다.

Str(char \*neyong) : neyong의 string 길이+1의 크기를 가지는 str을 생성하고 복사

~Str : 프로그램이 종료되기 전 소멸자로 메모리를 해제

int length(void) : str의 string 길이를 값 len 을 리턴한다.

char \*contents(void) : str의 주소를 리턴한다.

int compare(class Str& a) : class a를 받아 a.contents와 str을 비교한다.

int compare(char \*a) : 문자열 a와 str을 비교한다.

void operator=(char \*a) : = 연산자 재정의

1. 기존 str의 메모리를 해제한다.

2. a의 string 길이+1 만큼의 크기를 가진 배열 str을 새로 생성한다.

3. a의 내용을 str에 복사한다.

void operator=(class Str& a) : = 연산자 재정의

1. class a를 받고, a.contents의 내용을 str에 복사해야한다.

2. this 포인터와 재정의한 = 연산자를 이용한다.

Str의 알고리즘은 다음과 같다.

1. Str로 객체를 생성한다.

1-1. 배열의 크기를 받는다. Str(int leng)으로 간다. 이때 내용은 비어있다.

1-2. string의 내용을 받는다. Str(char \*neyong)으로 간다.

2. 객체가 담고 있는 string의 길이는 length에 접근해 알 수 있다.

3. 객체가 담고 있는 string의 내용은 contents에 접근해 알 수 있다.

4. 객체의 string과 비교하고싶은 문자열 a는 compare로 비교한다.

5. 객체끼리의 비교는 객체 안의 contents값에 접근해 compare로 비교한다.

6. 객체에 새로운 string을 할당할 때, 새롭게 정의한 연산자 =을 사용한다.

7. 객체에 다른 객체의 string을 할당할 때 역시 새롭게 정의한 연산자 =를 사용한다.

위 프로그램의 공간 복잡도는 주어지는 string의 길이 len와 같다. compare 함수의 시간복잡도 Big-‘oh’는 string의 길이가 len 이므로 최소 O(len)이다. 객체에 string을 복사하는 것 역시 string의 길이가 len이기 때문에, 역시 최소 O(len)이다. 또한, 위 프로그램은 warining이 뜰 수 있는데 이는, compare 함수나 Str(char \*neyong)은 인자를 char \*로 받지만, main에서는 const char로 넘겨주고있기 떄문이다.