

CSE3013 (컴퓨터공학 설계 및 실험 I)

CPP-1 결과 보고서

서강대학교 컴퓨터공학과 박수현 (20181634)

서강대학교 컴퓨터공학과

1 목적

실습 시간에 작성한 프로그램의 알고리즘과 자료 구조를 요약하여 기술한다.

2 문제

2.1 RangeArray

`class Array` 이 자료구조의 공간 복잡도는 $\mathcal{O}(size)$ 이다.

- `int *data` (protected): 런타임에 메모리를 할당하여 자료가 저장되는 변수이다.
- `int len` (protected): 배열의 크기를 저장하는 변수이다.
- `Array(int size)` (public): 크기가 `size`인 배열을 생성한다. 시간 복잡도는 $\mathcal{O}(size)$ 이다.
- `~Array()` (public): 배열을 `delete[]`로 메모리에서 제거한다. `Array` 안의 원소는 시간 복잡도는 기본 자료형인 `int`이므로, 시간 복잡도는 $\mathcal{O}(1)$ 이다.
- `int length()` (public): 배열의 크기를 반환한다. 시간 복잡도는 $\mathcal{O}(size)$ 이다.
- `int& operator[](int i)` (public): $0 \leq i < len$ 일 경우, 배열의 `i`번째 위치의 주소를 반환한다. 메모리의 특정 위치를 액세스하는 시간 복잡도는 $\mathcal{O}(size)$ 이다.
- `int operator[](int i) const` (public): $0 \leq i < len$ 일 경우, 배열의 `i`번째 위치의 값을 반환한다. 메모리의 특정 위치를 액세스하는 시간 복잡도는 $\mathcal{O}(size)$ 이다.
- `void print()` (public): 배열의 모든 원소를 출력한다. 시간 복잡도는 $\mathcal{O}(size)$ 이다.

`class RangeArray : public Array` 이 자료구조의 공간 복잡도는 $\mathcal{O}(high - low + 1)$ 이다.

- `int low` (protected): 배열의 시작 인덱스를 지정한다.
- `int high` (protected): 배열의 끝 인덱스를 지정한다.

- `RangeArray(int low, int high)` (public): 크기가 $high - low + 1$ 인 배열을 생성한다. 이 메서드는 `Array` 생성자를 상속받는데, 따라서 시간 복잡도는 `Array`를 만드는 시간 복잡도 $\mathcal{O}(high - low + 1)$ 와 같다.
- `~RangeArray()` (public): 배열을 메모리에서 제거한다. 실제로는 `~Array()`가 실행되게 된다. 시간 복잡도는 $\mathcal{O}(1)$ 이다.
- `int baseValue()` (public): 배열의 시작 인덱스를 반환한다. 시간 복잡도는 $\mathcal{O}(1)$ 이다.
- `int endValue()` (public): 배열의 끝 인덱스를 반환한다. 시간 복잡도는 $\mathcal{O}(1)$ 이다.
- `int& operator[](int i)` (public): $low \leq i \leq high$ 일 경우, 배열의 i 번째 위치의 주소를 반환한다. 실제로는 `Array::operator[](i + left)`가 실행되게 되며 시간 복잡도는 $\mathcal{O}(1)$ 이다.
- `int operator[](int i) const` (public): $low \leq i \leq high$ 일 경우, 배열의 i 번째 위치의 값을 반환한다. 실제로는 `Array::operator[](i + left)`가 실행되게 되며 시간 복잡도는 $\mathcal{O}(1)$ 이다.

2.2 Str

`class Str` 이 자료구조의 공간 복잡도는 $\mathcal{O}(len + 1)$ 이다.

- `char *str` (protected): 문자열의 정보를 Null-terminated string으로 저장한다.
- `int len` (protected): 문자열의 크기를 저장한다.
- `Str(int length)` (public): 크기가 `length`인 문자열을 생성한다. Null-terminated string으로 저장하기 위해 `str`에는 `length + 1`만큼의 원소를 메모리에 할당해 준다. 시간 복잡도는 $\mathcal{O}(length + 1)$ 이다.
- `Str(char *contents)` (public): 내용이 `contents`인 문자열을 생성한다. $|contents| + 1$ 만큼의 원소를 메모리에 할당해 준 후 `strcpy`를 사용한다. 시간 복잡도는 $\mathcal{O}(|contents| + 1)$ 이다.
- `~Str()` (public): `str`을 메모리에서 해제한다. 시간 복잡도는 $\mathcal{O}(length + 1)$ 이다.
- `int length()` (public): 문자열의 크기 `len`을 반환한다. 시간 복잡도는 $\mathcal{O}(1)$ 이다.
- `char *contents()` (public): 저장되어 있는 문자열 `str`을 반환한다. 시간 복잡도는 $\mathcal{O}(1)$ 이다.
- `int compare(char *a)` (public): `strcmp`를 이용해 `str`와 `a`를 비교한 결과를 반환한다. 시간 복잡도는 $\min(len, |a|)$ 이다.
- `int compare(Str &a)` (public): `(*this).compare(a.contents())`을 수행한다. 시간 복잡도는 $\min(len, |a|)$ 이다.
- `void operator=(char *a)` (public): `strcpy`를 이용해 `str`에 `a`를 복사한다. `a`가 `nullptr`일 경우 빈 문자열을 저장한다. `len`에는 `a`의 길이를 저장하며, `str`의 크기가 `a`의 길이보다 작을 경우 `a`의 길이만큼 큰 새로운 배열을 만들어 복사한다. 시간 복잡도는 $\min(|a|)$ 이다.
- `void operator=(Str &a)` (public): `(*this) = a.contents()`을 수행한다. 시간 복잡도는 $\min(|a|)$ 이다.