

**자료구조론**

과제 HW#2

보고서 작성 서약서

1. 나는 타학생의 보고서를 베끼거나 여러 보고서의 내용을 짜집기하지 않겠습니다.

2. 나는 보고서의 주요 내용을 인터넷사이트 등을 통해 얻지 않겠습니다.

3. 나는 보고서의 내용을 조작하지 않겠습니다.

4. 나는 보고서 작성에 참고한 문헌의 출처를 밝히겠습니다.

5. 나는 나의 보고서를 제출 전에 타학생에게 보여주지 않겠습니다.

나는 보고서 작성시 윤리에 어긋난 행동을 하지 않고 정보통신공학인으로서 나의 명예를 지킬 것을 맹세합니다.

2020년 7월 17일

학부 정보통신공학과

학년 3

성명 김현조

학번 12181761



1. 개요

이중 연결 리스트 (Doubly Linked List)로 List ADT와 Sequence ADT를 구현한다.

그 후에 버블 정렬 (Bubble Sort)를 통해서 알고리즘의 실행시간을 분석하고

선형 탐색 (Linear Searching) 함수를 구현하여 프로그램을 설계한다.

2. 구현 방법 – 자세한 구현은 주석을 이용하였다.

List ADT란 linear 구조이고 제약이 없기 때문에 임의의 위치에 모두 연산이 가능하다.

Position의 시퀀스므로 연속 공간이 아니여도 가능하다.

Node에는 trailer와 header 2개의 dummy node가 있으므로 양방향을 위해 이중 연결 리스트(DLL)로 List ADT를 구현한다.

Node와 NodeList Iterator 등 편의성을 위해서 독립적인 class로 만들어주었다.

Sequence ADT는 Vector ADT와 List ADT의 합집합이다.

Vector ADT는 index 기반으로, List ADT는 position기반이기 때문에

Sequence ADT를 구현할 때는 List ADT 전체를 상속 받고, Vector ADT 일부분 연산을 추가적으로 멤버함수를 받는다.

정렬 sorting은 순서화되지 않은 원소들을 가진 sequence를 재순서화, 정렬시키는 것이다.

그 중에서 버블 정렬은 sequence에 있는 data 순서가 맞는지 틀리는지에 따라서 옆 데이터와 순서를 계속 바꾼다.

배열 기반으로 버블 정렬을 구현하면 실행시간은 Big-Oh식으로 O(n^3)이지만,

List 기반으로 버블 정렬을 구현하면 실행시간은 O(n^2)로 줄어든다.

Node를 통해서 List ADT와 Sequence ADT 구현을 성공하고,

또한 버블 정렬을 이용하여 실행시간이 짧아짐을 확인하였는데

sequence에서 특정 key 값 k를 찾는 ‘선형 탐색(linear searching)’ 함수를 구현하는 것에 문법적인 에러가 생겼다.

다만, 주어진 main문 마지막에 있는 erase(0)에 에러가 발생하여 erase(ns.begin())으로 수정하였다. Position 정보를 받아야 하는데 erase(0)의 0을 nullptr로 인식하고 받아주는 것을 확인하여 가르치는 지점이 잘못되었기 때문에 begin()을 이용하였다.