[자료구조와실습 9주차]



|  |  |
| --- | --- |
| **■과 목 명** | 자료구조와실습 |
| **■담당교수** | 김시명 교수님 |
| **■제 출 일** | 2023.10.30 |
| **■학 과** | 컴퓨터공학전공 |
| **■ 학 번** | 2020112736 |
| **■성 명** | 안성현 |

1. 전위 연산과 후위 연산에 대해 서술하고 Stack에서 어떻게 쓰이는지 서술하시오.

A. 전위 연산은 변수에 대해 먼저 연산해주고 할당, 후위 연산은 먼저 할당해주고 연산하는 것이다. 전위 연산은 semicolon 이전에 연산하고, 후위 연산은 semicolon 이후에 연산한다고 볼 수 있다. 이는 Stack을 구현할 때, Push와 Pop에서 쓰이는데, Push에서는 top의 값을 먼저 올리고 그 인덱스에 해당하는 공간에 값을 넣어줘야 하기 때문에 전위 연산을 사용하고, Pop에서는 top 인덱스에 해당하는 공간에 값을 삭제하고 top의 값을 내려줘야 하기 때문에 후위 연산을 사용한다.

2. front, rear연산의 측면에서 일반 Queue와 Circular Queue의 차이점을 서술하시오.

A. 일반 queue는 front, rear가 마지막 인덱스를 넘어서면 따로 재조정을 해주어야 한다는 단점이 있다. 이 단점을 해결하기 위해 나온 Circular Queue는 front와 rear에 대해 modular 연산을 통해 마지막 인덱스를 넘어서면 0으로 되돌아가게 한다.

3. ADT (Abstract Data Type)에 대해 간략히 서술하고 ADT의 성능을 평가하는 기준 3가지를 서술하시오.

A. ADT는 데이터 객체와 연산으로 이루어진 Data Type를 추상화한 형식을 말한다. 데이터 객체와 연산이 무엇인지는 정의가 되지만, 이에 관한 구현은 드러나지 않아 캡슐화되어 있다는 특징이 있다. ADT의 성능 평가 기준에는 가독성, 메모리 공간 효율성, 실행시간 등이 있다.

4. 다음 행렬을 희소행렬로 바꾸시오.

A.

5. 연결 리스트의 삽입, 삭제 함수를 구현하시오.

- 삽입 : void insert(list\_node\* p, list\_node\* new) : p뒤에 new 삽입

- 삭제 : void delete(list\_node\* p, list\_node\* node) : p 뒤에 node 삭제

void insert(list\_node\* p, list\_node\* new) {

if (p) {

new -> link = p -> link;

p -> link = new;

}

else {

new -> link = head;

head = new;

}

}

void delete(list\_node\* p, list\_node\* node) {

if (p) {

p -> link = node -> link;

}

else {

head = head -> link;

}

free(node);

}