

자료구조 (큐)

1. 다음 중 큐와 관계가 없는 것은

- 1) rear 2) front 3) top 4) FIFO

2. 테크를 컴퓨터 내부에 나타내려고 할 때 다음 중 가장 알맞지 않은 자료구조는 ?

- 1) 밀집 리스트(dense list) 2) 순차적 리스트 3) 단순 연결리스트 4) 이중 연결리스트

3. 다음 중 테크에 대한 옳은 설명을 모두 나타낸 것은?

- 1) 스택과 큐의 복합체이다

자) 리스트의 양 끝에서 삽입(입력)과 삭제(출력)가 모두 일어날 수 있다.

- 3) 테크의 표현 방법에는 스택을 이용하는 방법, 1차원 배열을 이용하는 방법, 연결 리스트를 이용한 방법 등이 있다

- 4) 경우에 따라서는 삽입(입력)이나 삭제(출력)가 한쪽 끝에서만 일어나도록 테크를 제한할 수 있다

4. 프로그램 작업들이 SPOOL('simultaneous peripheral operation online)

지역 내부에서 CPU의 할당을 기다릴 때 사용되는 자료구조는?

- 1) 스택 2) 큐 3) 테크 4) 트리

5. 다음 중 큐의 행동방식을 나타내는 것은

- 1)LIFO 2)FIFO 3) top-down 4) bottom-up

6. 다음 중 큐에 새로운 노드가 삽입될 때의 연산을 옳게 나타낸 것은

- 1) front 의 값을 하나 증가시킨다

- 2) front의 값을 하나 감소시킨다

- 3) rear의 값을 하나 증가시킨다

- 4) rear의 값을 하나 감소시킨다.

7. 운영체제의 작업 스케줄링 등에 응용되는 것으로 가장 적합한 자료구조는

- 1) 스택 2) 큐 3) 연결리스트 4) 트리

8. 크기가 n인 큐의 front의 값을 f, rear의 값을 r이라고 하자. 다음 중 큐에서 오버플로우가 발생하는 경우는

- 1) $r=n$ 2) $f \geq n$ 3) $r=0$ 4) $f=0$

9. 큐가 1차원 배열 $Q[n]$ 으로 주어졌을 때, 다음 동작들을 모두 수행한 다음 큐 Q의 상태는? 단, 초기조건은 $front=rear=-1$

- 1) 처음에는 빈 큐 2) 노드 A를 삽입 3) 노드 B를 삽입 4) 노드 C를 삽입 5) 삭제 6) 삭제

10. 나머지 (modulo)연산자가 사용되는 자료구조는?

1) 원형 큐 2) 스택 3) 데크 4) 우선 순위 큐(priority queue)

11. 다음과 같은 원형 큐에 대하여 다음의 동작들을 한 다음 큐의 상태는 front=0, rear=3

0 1 2 3 4

	A	B	C			
--	---	---	---	--	--	--

1) 노드 D를 삽입 2) 노드 E를 삽입 3) 두 개의 노드들을 삭제 4) 노드 F를 삽입
5) 노드 G를 삽입 6) 삭제

12 위의 문제에서 모든 동작들을 수행한 다음 front와 rear의 값은?

13. 다음과 같은 데크에 대하여 다음의 동작들을 한 다음 데크의 상태는?

left =1, right =3

0 1 2 3 4

	A	B	C			
--	---	---	---	--	--	--

1) 왼쪽 끝에 노드 D를 삽입 2) 오른쪽 끝에 세 개의 노드 E, F, G를 삽입
3) 왼쪽 끝에서 두 개의 노드들을 삭제 4) 왼쪽 끝에 노드 F를 삽입 5) 오른쪽 끝에 노드 H를 삽입

14. 위 문제에서 모든 동작들을 수행한 다음 left 와 right의 값은?

15. 다음은 큐의 삭제 알고리즘이다 빈 칸의 내용으로 옳은 것은?

```
DELQ(item, Q, front, rear)
    if front=rear then call Q_EMPTY
    (          )
    item←Q(front)
end DELQ
```

1) front←front-1 2)front←front+1 3) rear←rear-1 4) rear←rear+1

16. 원형 큐의 삽입 알고리즘이다. 빈칸에 맞는 것은? (단, 크기는 n이다)

```
ADDCQ(item, Q, front, rear)
    (          ⊖          )
    if front=rear then call (          ⊕          )
    Q(rear)←item
end ADDCQ
```

17. 다음 알고리즘을 무엇을 수행하는 알고리즘인가?

```

PROCEDURE AA
    IF front = rear THEN CALL EMPTY
    front ← (front + 1) MOD n
    item ← POOL(front)
END AA

```

1) 큐 delete 2) 원형 큐 delete 3) 큐 ADD 4) 원형 큐 ADD

18. 원형 큐를 사용하는 가장 큰 이유는?

- 1) 선형 큐는 새로운 데이터의 첨가와 삭제가 어렵기 때문
- 2) 선형 큐는 데이터 검색에서 많은 시간이 소비되는데 이를 방지하기 위해서
- 3) 선형 큐에서 Queue-Full 신호가 발생하면 큐에서 많은 데이터의 이동이 일어나는데 이를 방지하기 위해서
- 4) 원형 큐를 사용하면 Queue-Full이 발생하지 않는다.

18. 환형 큐에서 자료가 삽입되는 알고리즘은 어떤 순서로 나열해야 맞는가? 맞는 순서를 써라

- 1) rear = ++rear % MAX
- 2) queue[rear]=data
- 3) if(rear+1) %MAX==front overflow()

```

enQ(q,item)
    rear=(rear+1) mod n;
    if(front=rear) then QFull();
    q[rear]=item;

```

19. 다음의 선형 큐를 운영할 때 삽입하기 위한 프로그램이다. 순서가 맞는 것은

```

enter_queue(int data) {
    if(rear==MAX)-----(1)
        overflow_routine()
    rear=rear+1 -----(2)
    queue[rear]=data -----(3)
}

```

```

enQ(q,item)
    if(rear=n-1) QFull();
    rear=rear+1;
    q[rear]=item;

```

20. 크기가 5인 선형 큐에서 다음의 연산을 수행한다. 큐가 포화 상태가 되어 더 이상 작업을 할 수 없게 되는 시점을 설명하라.

A 삽입 -> B 삽입 -> 삭제 -> C 삽입 -> 삭제 -> 삭제 -> D 삽입 -> E 삽입 -> 삭제 -> F 삽입 -> 삭제