

[6장 연습문제]

01. 자료구조에 대한 설명으로 옳지 않은 것은?

- ① 스택은 Last-In-First-Out 처리를 수행한다.
- ② 큐는 First-In-First-Out 처리를 수행한다.
- ③ 스택은 서브루틴 호출, 인터럽트 처리, 수식 계산 및 수식 표기법에 응용된다.
- ④ 큐는 비선형 구조에 해당한다.

02. 다음 중 큐에 대한 설명으로 옳은 것은?

- ① 큐의 크기는 항상 미리 정해져 있어야 한다.
- ② 자료의 삽입과 삭제는 같은 위치인 끝에서 일어난다.
- ③ 자료의 입출력은 LIFO(Last-In-First-Out) 순서로 일어난다.
- ④ 자료의 삽입과 삭제는 모두 $O(1)$ 시간에 수행된다.

03. 큐에 대한 설명으로 틀린 것은?

- ① 자료의 삽입과 삭제가 Top에서 이루어진다.
- ② FIFO 방식으로 처리한다.
- ③ Front와 Rear의 포인터 두 개를 갖고 있다.
- ④ 운영체제의 작업 스케줄링에 사용된다.

04. 다음 설명과 일치하는 자료구조를 각각 바르게 연결한 것은?

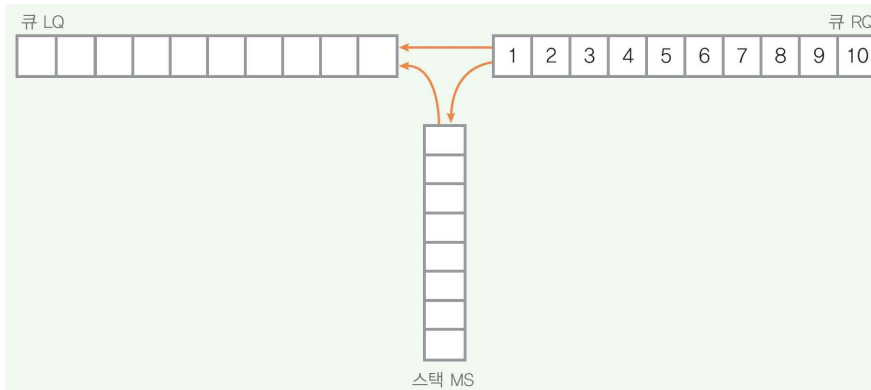
- (가) 먼저 추가된 항목이 먼저 제거된다.
- (나) 먼저 추가된 항목이 나중에 제거된다.
- (다) 항목이 추가된 순서에 상관없이 제거된다.

- | | | |
|------|--------|--------|
| (가) | (나) | (다) |
| ① 큐 | 연결 리스트 | 스택 |
| ② 스택 | 연결 리스트 | 큐 |
| ③ 스택 | 큐 | 연결 리스트 |
| ④ 큐 | 스택 | 연결 리스트 |

05. 다음 중 큐가 요구되는 작업으로 가장 적합한 것은?

- | | |
|-------------|--------------------|
| ① 작업 스케줄링 | ② 중위 표기식의 후위 표기 변화 |
| ③ 함수 호출과 리턴 | ④ 이진 트리의 중위 순회 |

06. 스택 수(Stack Number)란 다음 장치의 왼쪽 큐 LQ에 들어갈 수 있는 숫자열을 말한다. 오른쪽 큐 RQ로부터 데이터가 하나씩 중간에 있는 스택 MS를 거치거나 바로 LQ로 입력될 수 있다. 즉, 데이터가 LQ에 입력될 때 RQ에서 직접 입력되거나 MS에서 한 데이터를 삭제하여 LQ에 추가할 수 있다. 다음 장치에서 생성할 수 있는 스택 수는?



- | | |
|------------------------|------------------------|
| ① 1 8 5 6 7 4 9 3 10 2 | ② 3 4 6 7 8 5 9 2 1 10 |
| ③ 3 4 8 5 6 7 2 9 1 10 | ④ 9 4 7 5 6 3 8 1 2 10 |

07. 데크에 대한 설명을 바르게 짚지은 것은?

- ① \neg, \perp
- ② \neg, \perp, \sqsubset
- ③ \neg, \sqsubset, \equiv
- ④ \neg, \equiv

08 다음과 같은 원형 큐에 대해 '가'에서 '바'까지 연산을 차례로 수행했을 때, 수행이 완료된 후 큐의 상태는? (단, 현재 상태에서 $front = 0$, $rear = 2$ 이며, $front$ 에서는 삭제, $rear$ 에서는 삽입이 일어난다.)

[0]	[1]	[2]	[3]	[4]	[5]
	A	B			

- 가. 원소 C 삽입
나. 원소 두 개 삭제
다. 원소 한 개 삭제
라. 원소 D 삽입
마. 원소 E 삽입
바. 원소 F 삽입

- ①

[0]	[1]	[2]	[3]	[4]	[5]
F				D	E

②

[0]	[1]	[2]	[3]	[4]	[5]
F			C	D	E

③

[0]	[1]	[2]	[3]	[4]	[5]
	A	B	F		

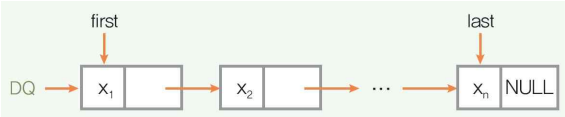
④

[0]	[1]	[2]	[3]	[4]	[5]
			F	D	E

09. 데크에 대한 설명으로 옳지 않은 것은?

- ① 입력 제한 테크는 Shelf이고 출력 제한 테크는 Scroll이다.
- ② 삽입과 삭제가 리스트의 양쪽 끝에서 발생할 수 있는 자료구조이다.
- ③ 스택과 큐의 장점으로 구성한 것이다.
- ④ Double Ended Queue의 약자이다.

10. 데크는 삽입과 삭제가 양끝에서 임의로 수행되는 자료구조이다. 다음 그림과 같이 단순 연결 리스트로 데크를 구현한다고 할 때 $O(1)$ 시간 내에 수행할 수 없는 연산은? (단, first와 last는 각각 데크의 첫 번째 원소와 마지막 원소를 가리키며, 연산이 수행된 후에도 데크의 원형이 유지되어야 한다.)



- ① insertFirst 연산 : 데크의 첫 번째 원소로 삽입
- ② insertLast 연산 : 데크의 마지막 원소로 삽입
- ③ deleteFirst 연산 : 데크의 첫 번째 원소를 삭제
- ④ deleteLast 연산 : 데크의 마지막 원소를 삭제

11. 스택 S와 원형 큐 Q의 초기 배열 상태가 다음과 같다고 가정하자. 여기서 T는 스택의 top을, R과 F는 큐의 rear와 front를 각각 나타낸다. (단, 스택에 관한 클래스 함수 push()는 스택에 삽입, pop()은 스택의 top이 가리키는 값을 삭제, top()은 스택의 top이 가리키는 값을 반환한다. 큐에 관한 클래스 함수 front()는 큐의 front가 가리키는 값의 반환, remove()는 큐의 front가 가리키는 값의 삭제를 나타낸다.) 오른쪽 문장이 순서대로 실행된 후 Q의 상태는?

S

T

3

6

Q

R

2

6

F

7

4

```

S.push(Q.front( ));
S.push(S.top( ));
S.pop( );
Q.remove( );
S.push(Q.front( ));

```

***보기 오탈자 수정: 빨간색 내용 추가**

① S

3

6

Q

2

6

7

4

② S

3

6

7

Q

2

6

4

③ S

3

6

7

4

Q

2

6

7

4

④ S

3

6

7

4

Q

2

6

4

12. 택시 정거장에서 줄을 서서 순서대로 택시를 타는 것과 유사한 자료구조는?

- ① 스택 ② 큐
③ 트리 ④ 그래프

13. 일상생활에서 발견할 수 있는 큐의 예를 설명하시오.

14. 큐와 스택의 구조를 비교하여 설명하시오.

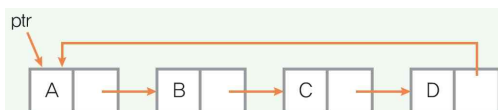
15. 1차원 배열의 선형 큐에서 잘못된 포화 상태 문제를 해결하는 방법을 설명하시오.

16. 크기가 5인 선형 큐에서 다음 연산을 수행한다. 큐가 포화 상태가 되어 더 이상 작업을 할 수 없게 되는 시점을 설명하시오.

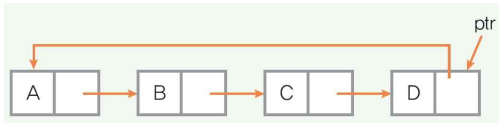
A 삽입 → B 삽입 → 삭제 → C 삽입 → 삭제 → 삭제 → D 삽입 → E 삽입 → 삭제 → F 삽입 → 삭제

17. 큐를 연결 리스트로 구현하는 경우 일반적으로 front와 rear 등 포인터를 두 개 사용한다. 만약 연결 리스트의 포인터를 하나밖에 사용할 수 없다면 가장 효율적인 구조는? (단, 큐에 A, B, C, D 순서로 입력되었고, 역원형 리스트는 데이터가 입력된 순서의 역순으로 만든 원형 리스트를 의미한다고 가정한다.)

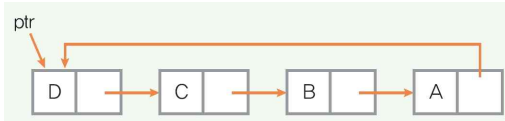
① 포인터가 첫 노드를 가리키는 원형 리스트 구조



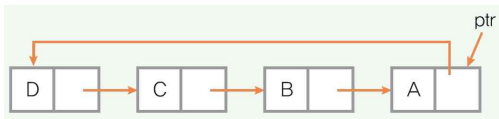
② 포인터가 마지막 노드를 가리키는 원형 리스트 구조



③ 포인터가 첫 노드를 가리키는 역원형 리스트 구조



④ 포인터가 마지막 노드를 가리키는 역원형 리스트 구조



18. 원형 큐에서 포화 상태와 공백 상태의 조건을 설명하시오.