

기말고사 예상 4지선다형 문제(2)

1. 다음 중 스택을 활용하는 대표적인 응용 분야로 가장 적절하지 않은 것은? (**d**)
a) 재귀 함수 호출 관리 b) 웹 브라우저의 뒤로 가기/앞으로 가기 기능 구현 c) 괄호 짝 맞추기 유효성 검사 d) 너비 우선 탐색(BFS) 구현 - 큐
정답: d) 너비 우선 탐색(BFS)은 큐(Queue)를 사용함.

2. 원형 배열로 구현된 큐에서 front와 rear 포인터만 사용할 때, 큐가 꽉 찼는지 빈지를 구분하기 위해 보통 한 칸을 비워둔다.
이 때 최대 저장 가능한 원소 개수는? (배열 크기 n) (**B**)
A. n
B. n-1
C. n+1
D. n/2

3. 큐(Queue)의 기본 동작 방식은 무엇인가? (**B**)
A. LIFO
B. FIFO
C. FILO
D. LILO

4. 큐에서 원소를 제거하는 연산은 무엇인가? (**C**)
A. push
B. pop
C. dequeue
D. peek

5. 원형 큐(Circular Queue)의 장점은? (**E**)
A. 메모리 낭비가 줄어든다 - 선형 큐
B. 코드가 단순하다
C. 항상 오버플로가 발생한다
D. 포인터를 사용하지 않는다
E. 원소의 재배치가 필요없다

6. 큐가 비었는지 확인할 때 사용하는 함수는? (**C**)
A. isFull()
B. isZero()
C. isEmpty()
D. isNull()

7. 큐가 가득 찼을 때 삽입 시도 시 나타나는 현상은?(B)

- A. Underflow
- B. Overflow
- C. Runtime Error
- D. Compile Error

8. 다음 중 큐를 이용하는 구조는?(B)

- A. 웹 브라우저 뒤로가기 //스택
- B. CPU 스케줄링(FCFS) = FIFO
- C. 실행 취소 기능 //스택
- D. 스택 기반 계산기//스택

9. 스택의 기본 동작 방식은?(B)

- A. FIFO
- B. LIFO (후입선출)
- C. FILO
- D. PRIORITY

10. 함수 호출(Call Stack)은 어떤 자료구조로 구현되는가?(B)

- A. 큐
- B. 스택
- C. 그래프
- D. 힙

11. push/pop이 일어나는 위치는?(C)

- A. front
- B. rear
- C. top
- D. root

12. 덱(Deque)의 특징은?(B)

- A. 한쪽에서만 삽입 가능
- B. 양쪽에서 삽입/삭제 가능
- C. 항상 오름차순 유지
- D. LIFO만 가능

13. 스택과 큐 모두의 기능을 가진 자료구조는?(B)

- A. 리스트
- B. 덱
- C. 배열
- D. 그래프

14. 다음 중 원형 큐(Circular Queue) 에서 $front == rear$ 인 상태에 대한 설명으로 가장 적절한 것은?

~~단, 구현 방식은 "front는 첫 번째 요소 바로 앞의 위치" 를 가리키는 방식이다.~~ (A)

- A. 큐가 비어 있다
- B. 큐가 가득 차 있다
- C. front가 rear의 다음 요소를 가리킨다
- D. 명확히 판단할 수 없다

15. 한 큐의 길이가 10이고, $front = 3$, $rear = 7$ (배열 인덱스 0~9)이라면 현재 큐에 들어있는 원소 수는? ($size = (rear - front + n) \% n$) (B)

(원형 큐 구현 가정)

$$(7-3+10)\%10 \\ 14\%10 = 4$$

- A. 3
- B. 4
- C. 5
- D. 6

16. 정렬된 연결 리스트로 우선순위 큐를 구현할 때 장점으로 가장 적절한 것은?(D)

- A. 삽입이 $O(1)$
- B. 삭제가 $O(n)$
- C. 삽입이 $O(n)$
- D. 삭제가 $O(1)$ *

17. 다음 중 데크(Deque)로 구현할 수 없는 자료구조는?(C)

- A. 스택
- B. 큐
- C. 우선순위 큐
- D. 양방향 큐

18. 배열을 사용하여 일반적인 선형 큐(Linear Queue)를 구현할 때, Dequeue 연산 후에 배열의 나머지 요소들을 앞으로 이동시키는(shifting) 과정이 필요하여 비효율적입니다. 이러한 요소 이동을 해결하고 공간 활용도를 높일 수 있는 큐 구현 방식은 무엇입니까?(D)

- A. 이중 연결 리스트 (Doubly Linked List) 기반 큐
- B. 데크 (Deque)
- C. 스택 2개로 구현한 큐
- D. 원형 큐 (Circular Queue) — 선형 큐 컴 보완

19. Deque를 (이중 연결 리스트(Doubly Linked List))로 구현할 경우, 프론트(Front)에 요소 삽입하는 연산과 리어(Rear)에서 요소 삭제하는 연산의 최악의 경우 시간 복잡도는 각각 무엇입니까? (A) Queue

- A. Insert Front: $O(\log N)$, Delete Rear: $O(1)$
- B. Insert Front: $O(1)$, Delete Rear: $O(N)$
- C. Insert Front: $O(1)$, Delete Rear: $O(1)$
- D. Insert Front: $O(N)$, Delete Rear: $O(N)$

20. 어떤 자료구조가 데크를 기반으로 구현되었는데, 오직 insert_front(앞에서 삽입)와 delete_rear(뒤에서 삭제) 연산만 허용된다고 가정합니다. 이 제한적인 자료구조는 다음 중 어떤 기본 자료구조의 특성을 모방합니까?()

- A. 힙 (Heap)
- B. 큐 (Queue)
- C. 스택 (Stack)
- D. 우선순위 큐 (Priority Queue)