

기말고사 예상 4지선다형 문제

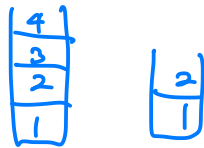
1. 다음 중 스택으로 구현하기 가장 부적합한 연산은? (D)

- A) 후위표기식 계산 *stack*
- B) 깊이 우선 탐색(DFS) *stack*
- C) 중위표기식 → 후위표기식 변환 *stack*
- D) 레벨 순회(Level-order traversal) *queue*

2. 다음 스택 동작 결과로 맞는 것은? (A)

Push: 1,2,3,4 수행 후 Pop 두 번 수행 → 남은 요소?

- A) [1,2]
- B) [1,2,3]
- C) [1,2,4]
- D) [3,4]



3. 스택 오버플로우(overflow)가 발생하는 원인으로 가장 적절한 것은? (B)

- A) Pop이 너무 많음
- B) Push로 용량 초과
- C) top을 잘못 감소함
- D) 동적 배열 미사용

→ 총재진 크기보다 push 多

4. 1~5 push 후 pop 3회, 다시 6 push. top의 값은? (A)

- A) 2
- B) 3
- C) 4
- D) 6

1 2 3 4 5 6
top -1 1 2

The diagram shows a sequence of numbers 1, 2, 3, 4, 5, 6. The numbers 3, 4, and 5 are crossed out with red lines. Below the sequence, the text 'top -1 1 2' is written, indicating the state of the stack after each operation: push 1 (top=-1), push 2 (top=1), pop 3 (top=1), pop 4 (top=1), pop 5 (top=1), push 6 (top=2).

5. 스택 기반 후위표기식 평가에서, 연산자가 등장하면 수행하는 행동은? (C)

- A) push
- B) pop 1회
- C) pop 2회 후 연산
- D) 아무것도 하지 않음

6. 스택에서 $top = -1$ 의 의미는? (C)

- A) overflow
- B) underflow
- C) 스택이 비어 있음
- D) 포인터 오류

7. 원형 큐(circular queue)에서 $front == rear$ 조건이 의미하는 상태는? (B)

- A) 큐가 가득 참
- B) 큐가 비어 있음
- C) 오류
- D) 메모리 부족

선형

8. 배열 기반 큐에서 front가 증가하면서 발생하는 비효율은? (B)

- A) 메모리 누수
- B) 뒤쪽 미사용 공간 증가
- C) pop 속도 저하
- D) push 속도 저하

9. 원형 큐에서 rear 위치 증가 연산은? (B)

- A) rear++
- B) rear = (rear + 1) % size
- C) rear = rear - 1
- D) rear = size - 1

10. 큐가 가득 찼는지 확인하는 조건은? (B)

- A) front == rear
- B) (rear+1) % size == front
- C) rear == size
- D) front == -1

11. 다음 큐 동작 결과로 맞는 것은?

enqueue: 1,2,3,4 후 dequeue 2회, enqueue 5

남아있는 순서? (A)

- A) 3,4,5
- B) 2,3,4
- C) 1,2,5
- D) 4,5

1 2 3 4 5

12. 큐에서 underflow가 발생하는 상황은? (B)

- A) rear overflow
- B) 비었는데 dequeue
- C) 가득 찼는데 enqueue
- D) 포인터 잘못 증가

13. 우선순위 큐로 구현하기 적절한 구조는? (A)

A) 작업 스케줄링 (우선처리)

B) DFS

C) Undo 기능

D) Round-robin

14. 우선순위 큐에서 꺼내는 요소의 특징은? (C)

A) 가장 먼저 들어온 요소

B) 임의 요소

C) 가장 높은 우선순위

D) 가장 낮은 우선순위

15. 데크의 가장 큰 특징은? (B)

A) 한쪽 끝 삽입

B) 양쪽 끝 삽입·삭제 가능

C) 우선순위 기반 삭제

D) FIFO 강제

16. 데크가 큐보다 유리한 상황은? (A)

A) 양쪽에서 데이터를 필요로 할 때

B) 단일 방향만 필요할 때

C) 우선순위를 관리할 때

D) 정렬이 필요할 때

17. 데크의 성질로 부적절한 것은? (C)

A) front와 back 모두 접근 가능

B) LIFO와 FIFO 모두 구현 가능

C) 중간 인덱스 접근 $O(1)$

D) 양쪽 끝에서 $O(1)$ 삭제

18. 데크가 다음 동작을 수행:

push_back 1,2,3 → pop_front 2회 → push_front 4

결과는? (B)

A) 3,4

B) 4,3

C) 1,4

D) 2,3

f 4, 2, 3 b

19. 데크를 구현할 때, 어떤 자료구조를 사용하는 것이 좋겠는가? (B)

A) 단순 연결 리스트 B) 이중연결리스트 C) 1차원 배열 D) 2차원 배열

20. 운영체제에서 프로세스 스케줄링을 위해 FIFO(First-In, First-Out) 원칙을 사용하여 요청을 관리해야 할 때, 가장 적합한 자료구조는 무엇입니까? (B)

A) 스택(Stack) B) 큐(Queue) C) 우선순위 큐(Priority Queue) D) 덱(Deque)

LIFO

FIFO