

<b>자료구조론</b>	<b>국가 전산 7급</b>	<b>2012년 7월 28일</b>
--------------	-----------------	---------------------

♣ 채용인원/합격선(16명/72.14점) - 양성평등.여자 69.28점 / 장애 57.14점 ♣

1. 다음 함수를 이용하여 recursive(4)를 수행한 결과 값은? [2012년 국가 7급]

```
int recursive( int n ) {
    if(n < 1) return 2;
    else return(22 * recursive(n-1) + 1);
}
```

- ① 149                      ② 265
- ③ 373                      ④ 597

♣ 재귀호출

- 
- recursive(0) = 2 → 완료조건
  - recursive(1) = 2<sup>2</sup> \* recursive(0) + 1 = 4 \* 2 + 1 = 9
  - recursive(2) = 2<sup>2</sup> \* recursive(1) + 1 = 4 \* 9 + 1 = 37
  - recursive(3) = 2<sup>2</sup> \* recursive(2) + 1 = 4 \* 37 + 1 = 149
  - recursive(4) = 2<sup>2</sup> \* recursive(3) + 1 = 4 \* 149 + 1 = **597**

정답 : ④

2. if 문장에서 포인터 변수 pt1과 pt2에 대한 설명으로 옳지 않은 것은? [2012년 국가 7급]

- ① if (&pt1==&pt2)의 경우에 서로 다른 포인터 변수가 동일한 주소에 있을 수 있다.
- ② if (pt1==pt2)는 같은 대상을 가리키는지를 검증한다.
- ③ if (\*pt1==\*pt2)의 경우에 가리키는 대상이 다를지라도 변수값은 같을 수 있다.
- ④ if (pt1==pt2)는 가리키는 대상의 주소가 동일한 것인지를 검증한다.

♣ 포인터 변수 pt1과 pt2

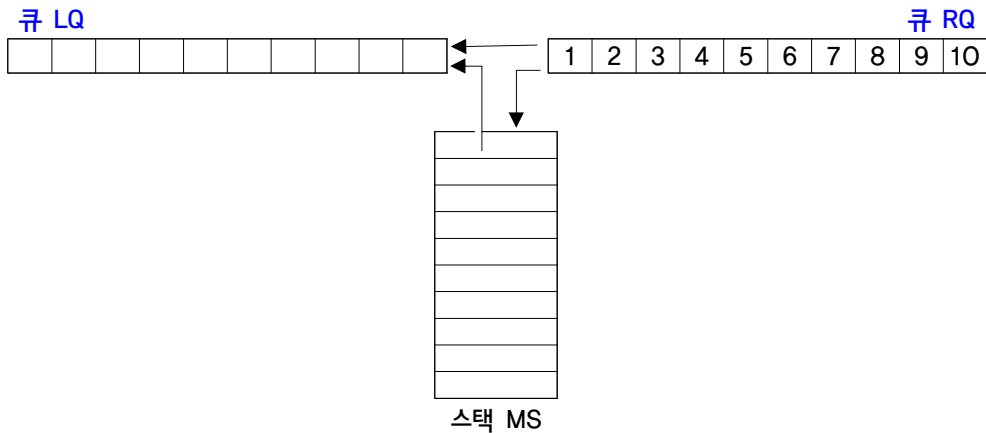
- 
- if (&pt1==&pt2)의 경우에 서로 다른 포인터 변수가 **동일한 주소에 있을 수 있다.(×)**  
 → 각 변수는 자신 고유의 메모리를 할당받는다.  
 → 해서, 서로 다른 포인터 변수가 **동일한 주소에 있을 수 없다.**

// 변수

- 먼저, 모든 변수는 각각 **고유의 메모리가 확보된다.**
- 포인터 변수도 변수이므로 각각 메모리가 확보된다.
- 해서, 서로 다른 포인터 변수가 동일한 메모리(주소)에 있을 수 없다.

정답 : ①

3. 스택 수(stack number)란 다음 장치의 왼쪽 큐 LQ에 들어 갈 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 삭제해서 LQ에 추가할 수 있다. 다음 장치에서 생성할 수 있는 스택 수는? [2012년 국가 7급]



- 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

☞ 스택 수(stack number) - 이상하게 출제 빈도가 높음(정확하게 출제된 경우)

push(1), push(2), LQ 입력(3), LQ 입력(4), push(5),  
 LQ 입력(6), LQ 입력(7), LQ 입력(8),  
 LQ 입력(pop(5)), LQ 입력(9), LQ 입력(pop(2)),  
 LQ 입력(pop(1)), LQ 입력(10)

정답 : ②

4. 합병정렬(merge sort)에 대한 설명으로 옳지 않은 것은? [2012년 국가직 7급]

- ① 점근분석(asymptotic analysis)할 때, 퀵정렬의 최악수행시간 보다 빠르다.
- ② 추가적인 메모리 사용없이 간단히 구현할 수 있다.
- ③ 크기  $n$ 인 입력에 대한 최악수행시간을  $T(n)$ 이라 할 때,  $T(n) = 2 \times T(n/2) + cn$ 으로 나타낼 수 있다.(단,  $c \geq 1$  상수)
- ④ 분할정복(divide and conquer) 기법을 사용한다.

☞ 합병정렬

- 통상적인 반복 합병정렬은 추가 메모리  $O(n)$ 이 필요하다.

정렬방법	수행시간	최악의 경우	기억공간	안정/불안정	제자리정렬
선택정렬	$O(n^2)$	$O(n^2)$	$n$	불안정	○
거품정렬	$O(n^2)$	$O(n^2)$	$n$	안정적	○
삽입정렬	$O(n^2)$	$O(n^2)$	$n$	안정적	○
셀정렬	$O(n^2)$	$O(n^2)$	$n$	불안정	○
힙정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	$n$	불안정	○
합병정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	$2n$	안정적	×
퀵정렬	$O(n \log_2 n)$	$O(n^2)$	$n + \text{stack}$	불안정	○ / ×
기수정렬	$O(k(n+q))$	$O(k(n+q))$	$(n+1)q$	안정적	×

// 제자리정렬(in-place sort)

- 제자리정렬은 정렬 과정에서 추가 기억장소를 조금만 더 사용하는 알고리즘이다.
- 제자리정렬은 정렬 대상 자료 이외의 추가 기억장소가 상수 개를 넘지 않는 알고리즘이다.
- 제자리정렬은 정렬 대상 자료에 비해 무시할 정도의 기억장소를 더 사용하는 알고리즘이다.
- 제자리정렬 : 선택정렬, 버블정렬, 삽입정렬, 셀정렬, 힙정렬

// 합병정렬

- 기본적으로, 합병정렬은 제자리 정렬(in-place sort)이 아니다.
- 하지만, 합병정렬은 제자리 정렬로 구현할 수 있다.
- 합병정렬의 종류는 많다.
- 합병정렬은 반복 또는 순환으로 구현할 수 있다.

5. 방향그래프의 행(row)은 출발 노드로 열(column)은 도착 노드로 매핑하여 다음과 같이 인접행렬로 표현될 때, 위상정렬(topological sort)이 가능한 것은? [2012년 국가 7급]

①

	a	b	c	d	e	f
a		1	1			
b			1	1		
c				1	1	
d			1			1
e				1		1
f						

②

	a	b	c	d	e	f
a		1	1			
b			1	1		
c				1	1	
d						1
e		1		1		1
f						

③

	a	b	c	d	e	f
a		1	1			
b			1	1		
c				1	1	
d						1
e			1			1
f						

④

	a	b	c	d	e	f
a		1	1			
b			1	1		
c				1	1	
d						1
e			1			1
f	1					

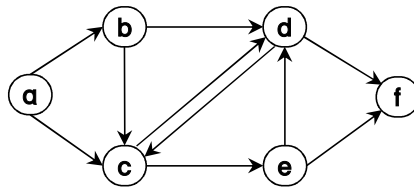
☞ 위상정렬 - 정점 작업 네트워크

• 비사이클 방향그래프이어야 위상정렬이 가능하다.(방향사이클이 없어야 한다)

①의 인접행렬표현에 대한 방향그래프

①

	a	b	c	d	e	f
a		1	1			
b			1	1		
c				1	1	
d			1			1
e				1		1
f						

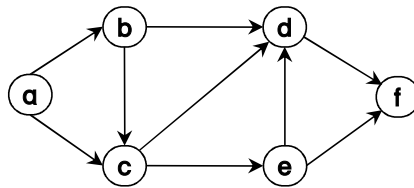


①은 방향사이클 존재

③의 인접행렬표현에 대한 방향그래프

③

	a	b	c	d	e	f
a		1	1			
b			1	1		
c				1	1	
d						1
e			1			1
f						



③은 방향사이클 없음

• 항목 ③은 방향사이클이 없으므로 위상정렬이 가능하다.