

2016 학년도 2 학기 중간고사 (1/2)	과	학 과	학 년	점 수
담당교수	목	학 번		
시험일시	명	성 명		

심 정 섭
10월 20일 목요일

자료구조

1. 다음 설명들 중 옳지 않은 것들을 모두 고르고, 옳지 않은 이유를 각각 50자 이내로 설명하시오.

- ① $\Omega(n)$ 에 속하는 함수는 모두 $\Omega(\log n)$ 에 속한다.
- ② 사칙연산(+, -, ×, ÷), 변수에 값 할당, 함수 호출 및 복귀는 모두 기본연산(primitive operation)이다.
- ③ 배열은 연속된 주소공간을 할당받아야 하므로 프로그램 수행중에는 생성될 수 없다.
- ④ 연결리스트의 노드로 사용되었던 공간은 프로그램 수행 중에 재사용될 수 있다.

답:

2. 다음 연산 또는 알고리즘들이 크기 n 인 데이터에 대해 수행되었을 때, 최악의 경우에 대한 점근분석 수행 시간이 가장 빠른 순서대로 나열하시오. 단, 수행시간이 같은 연산은 번호가 빠른 연산을 먼저 쓰시오.

- ① 스택을 배열로 구현했을 때, 검색연산
- ② 벡터를 환형배열(circular array)로 구현했을 때, i 번째 위치에 삽입연산
- ③ 정렬된 배열에서의 이진탐색(binary search)
- ④ 이중연결리스트(doubly linked list)의 테일(tail)에서의 삭제연산

답: () -> () -> () -> ()

3. 다음은 배열로 구현한 스택 추상자료형(ADT)의 top() 연산의 의사코드(pseudocode)이다. 이를 참조하여 push(e)와 pop() 연산(반환값 없음)의 의사코드를 작성하시오.

```

Algorithm top()
  if empty() then throw StackEmpty exception
  return  $S[t]$ 

```

```

Algorithm push( $e$ )

```

```

Algorithm pop()

```

4. 시퀀스(sequence) ADT를 각각 환형배열과 이중연결리스트로 구현하였다. 강의시간에 설명한 시퀀스 ADT의 연산들 중 최악수행시간의 복잡도가 아래 표와 같은 연산들을 하나씩만 선택하여 그 기능을 각각 30자 이내로 서술하시오.

연산	환형배열	이중연결리스트
(1)	$O(n)$	$O(n)$
(2)	$O(n)$	$O(1)$
(3)	$O(1)$	$O(n)$

(1)

(2)

(3)

5. 한 개의 정수와 크기 10인 문자형 배열로 구성된 데이터를 각각 이중연결리스트와 이진트리(binary tree)로 구현하려 한다. 빈 칸을 C, C++의 문법에 맞도록 채워 구조체 node를 정의하시오.

① 이중연결리스트

```

struct node {

```

```

}

```

② 이진트리

```

struct node {

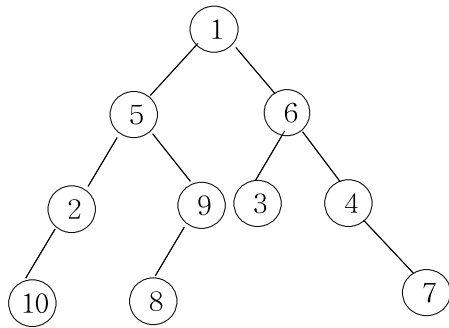
```

```

}

```

6. **강의시간에 설명한 방법으로** 아래 이진트리를 좌표평면에 그리려고 한다. 각 노드 안의 수가 노드번호라 하고 루트(root)인 ①번 노드의 좌표가 (6, 0)일 때, ②번 노드부터 ⑩번 노드들의 좌표를 쓰시오.

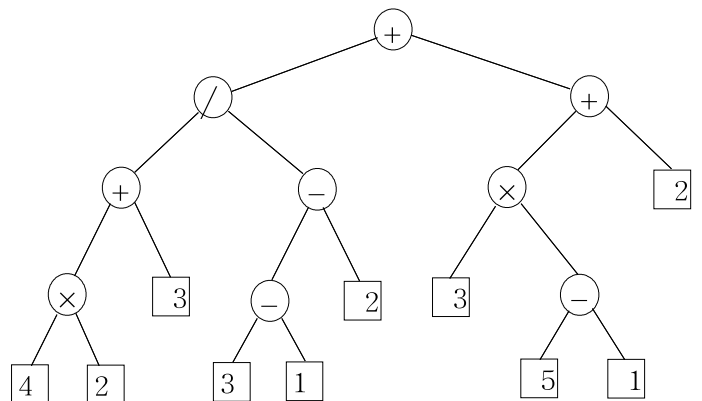


②: (,) ③: (,) ④: (,)
 ⑤: (,) ⑥: (,) ⑦: (,)
 ⑧: (,) ⑨: (,) ⑩: (,)

7. 노드 v 가 루트인 이진트리에 저장된 수식을 화면에 출력하는 `printExpression(v)` 함수의 의사코드를 작성하시오. 단, 화면에 문자열 `s`를 출력하는 함수는 `print(s)`이며, 이진트리의 노드에 사용할 변수는 어떤 역할을 하는지 쉽게 알 수 있도록 표현하시오.

8. n 개의 노드를 저장하고 있는 트리 T 를, **강의시간에 설명한 방법**으로 이진트리 T' 을 이용하여 구현하였다. T 의 노드 u 의 모든 형제노드(sibling)들의 데이터를 출력하는 함수를 구현하려 한다. 방법과 시간복잡도에 대해 100자 내외로 설명하시오.

9. 아래 그림과 같이 이진트리에 데이터들이 저장되어 있을 때, 각각 다음 방식으로 순회하였을 때, 노드를 방문(visit)하는 순서대로 해당 노드의 데이터를 쓰시오.



(1) 전위순회(preorder traversal)

(2) 후위순회(postorder traversal)

(3) 중위순회(inorder traversal)