자료구조론 국가 전**산 7급** 2007년 8월 9일

☆ 채용인원/합격선(20명/73.57점) - 장애 69.28점 ☆

1. 크기가 10인 해시테이블을 배열을 이용하여 만든다고 가정하자. 해시함수로 h(k) = k mod 10을 사용한다. 여기서 mod는 모듈로(modulo) 함수를 의미한다. 데이터가 다음과 같은 순서로 입력된다고 할 때 발생하는 충돌 횟수의 총합으로 옳은 것은?(단, 배열의 인덱스는 [0]-[9]까지 이고, 처음에 배열은 비어 있으며, 충돌을 해결하기 위해서는 배열의 다음 빈 공간에 데이터를 입력하는 선형조사법(linear probe)을 사용한다) [2007년 전산 7급]

70 17 26 69 72 53 31 27 23 13

- ① 3
- (2) 4
- 3 5
- 4) 6

☆ 선형조사법 - 충돌 횟수

	0	1	2	3	4	5	6	7	8	9	→ 버킷주소
	70	31	72	53	23	13	26	17	27	69	
L				1	1			1			J
				23	13			27			
				13							

$h(70) = 70 \mod 10 = 0$	
$h(17) = 17 \mod 10 = 7$	
$h(26) = 26 \mod 10 = 6$	
$h(69) = 69 \mod 10 = 9$	
$h(72) = 72 \mod 10 = 2$	
$h(53) = 53 \mod 10 = 3$	
$h(31) = 31 \mod 10 = 1$	
$h(27) = 27 \mod 10 = 7$	충돌 1번 발생, 다음 빈 공간을 탐색하여 저장
$h(23) = 23 \mod 10 = 3$	충돌 1번 발생
$h(13) = 13 \mod 10 = 3$	충돌 1번 발생

• 충돌 횟수 = 4번

2. 최대합으로 우선순위큐를 구현하려 한다. 우선순위를 나타내는 9개의 데이터가 큐에 다음과 같은 순서대로 삽입되었다. 1개의 데이터가 큐에서 삭제된 후, 재 정렬된 합에서 가장 마지막 원소는 무엇인가?(단, 숫자가 클수록 우선순위가 높다고 가정한다) [2007년 전산 7급]

24 17 29 22 20 31 27 18 21

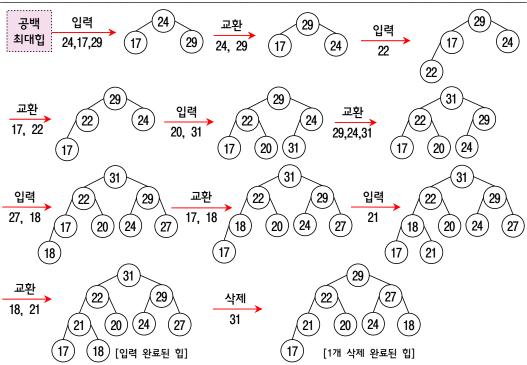
① 17

② 18

③ 20

4 21

☆ 자료를 차례로 입력하면서 힙구조 조건에 맞지 않으면 힙구조로 변경시킨다.



- 힙에서 자료 삭제는 먼저, 루트(31)를 삭제하고,
- •마지막 자료(18)를 루트에 넣고, 힙구조가 되도록 재 정렬한다.

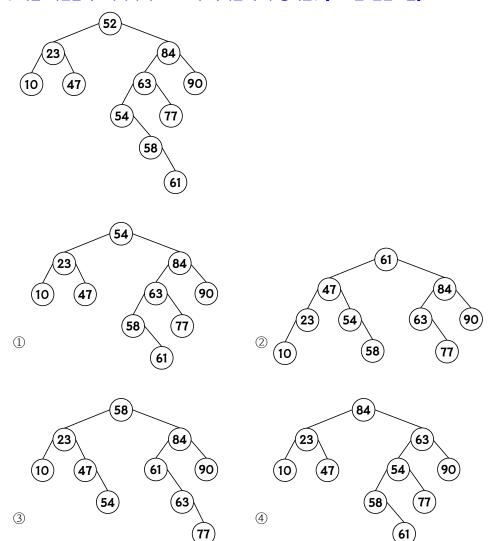
	•이 문제는 처음 발표된 '정답가안'에서 정답이 '② 18'로 발표되었다.
	•해서, 이의 신청을 하였고, 최종 답안은 '① 17'로 변경되었다.
이의 신청	• 2003년도 기술고시 '자료구조론' 시험에서도 이 유형의 문제에 이의 신청이 많았었다.
	•최종 정답은 위에서 설명한 원리대로 적용한 것만 인정되었다.
	• 힙구조 운영과 힙정렬에서 힙구조를 이용하는 원리는 다르다.
히고ㅈ	가근가 이추려디느 시저에 히 조거에서 버어나며 히그ス근 벼겨시키다

 합구조
 자료가 입출력되는 시점에 힙 조건에서 벗어나면 힙구조로 변경시킨다.

 합정렬
 정렬 대상 자료를 트리에 모두 입력한 후 힙구조로 변경시킨다.

320 http://cafe.daum.net/pass365(홍재연)

3. 다음 이진탐색트리에서 루트노드가 삭제된 후의 상태는? [2007년 전산 7급]



☆ 이진탐색트리에서 루트 삭제

//이진탐색트리에서 루트 삭제는 다음 2가지이다.

• 왼쪽에서 가장 큰 값을 루트에 대입 : 47을 루트에 대입

• 오른쪽에서 가장 작은 값을 루트에 대입 : 54를 루트에 대입(주어진 문제의 답)

정답 : ①

4. 이진탐색트리 T에서 k보다 큰 키들의 개수를 찾는 클래스 함수 numGreaterThan(T, k)을 작성한다고 가정하자. 트리에 포함된 모든 키들은 상이하고, 각 노드 x는 링크 필드인 left 및 right와 데이터 필드인 data를 가지며, 추가적으로 하나의 데이터 필드 numDesc를 포함한다. numDesc는 x를 루트(root)로 하는 부분 트리에 포함된 모든 키들의 개수로 초기화되어 있다. 아래 코드에서 []에 해당하는 문장은 무엇인가? [2007년 전산 7급]

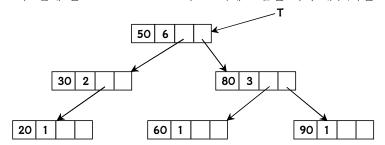
```
int numGreaterThan(T, k){
  if(T == NULL) return 0;
  else if(T.key <= k) return numGreaterThan(T.right, k);
  else return [ ];
}</pre>
```

- ① T.right.numDesc 1
- 2 numGreaterThan(T.left, k) + 1
- ③ T.right.numDesc + numGreaterThan(T.left, k) 1
- 4 T.right.numDesc + numGreaterThan(T.left, k) + 1

☆ 이진탐색트리

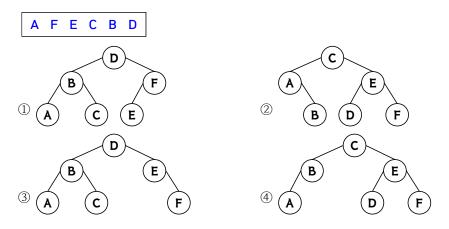
// 주어진 문제에 근거한 이진탐색트리 구조는 다음과 같다.

- 첫 번째 필드: data
- 두 번째 필드: numDesc 서브트리에 포함된 키의 개수(자신 포함)



```
// numGreaterThan(T, 65)인 경우 - 65보다 큰 키의 개수
numGreaterThan(T, 65) = numGreaterThan(T.right, 65)
= 1 + numGreaterThan(T.left, 65) + 1
= 1 + numGreaterThan(NULL, 65) + 1
= 1 + 0 + 1
= 2
```

5. 다음과 같은 순서로 입력되는 데이터에 대해 올바르게 AVL-트리를 구성한 것은? [2007년 전산 7급]



☆ AVL-트리

•자료 입력 순서에 따라 AVL-트리의 회전은 다음과 같다.

