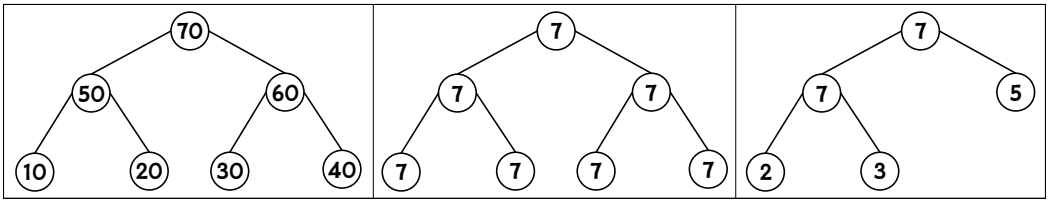
● 힙(heap)

힙은 완전이진트리의 특수한 형식으로 그 정의는 다음과 같다.

◈ 힙 정의

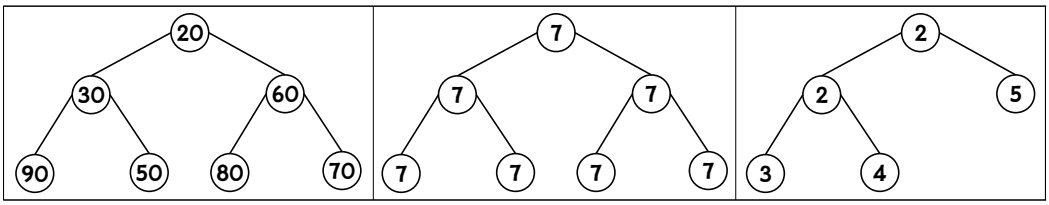
- **최대트리** : 부모노드 값 ≥ 자식노드 값(단, 자식이 있을 때)
- **최소트리** : 부모노드 값 ≤ 자식노드 값(단, 자식이 있을 때)
- **최대힙** : 최대트리이면서 완전이진트리이다.
- **최소힙** : 최소트리이면서 완전이진트리이다.

다음 3개의 트리는 모두 **최대힙**이다.(부모노드 값≥자식노드 값)



최대힙 1 최대힙 2 최대힙 3

다음 3개의 트리는 모두 **최소힙**이다.(부모노드 값≤자식노드 값)

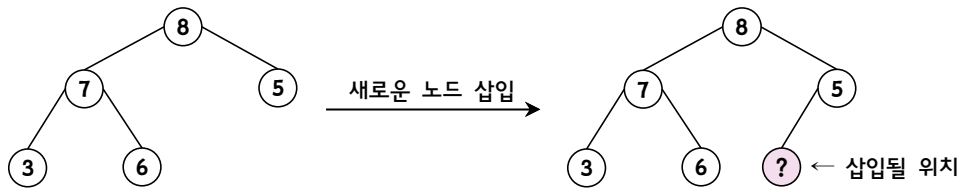


최소힙 1 최소힙 2 최소힙 3

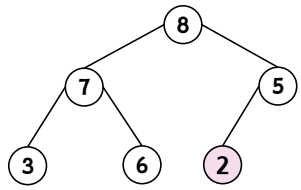
- 최대힙의 루트는 가장 큰 값을 가지고, 최소힙의 루트는 가장 작은 값을 가진다.
- 최대힙과 최소힙에서 루트 값을 제외하고는 다른 값들의 크기 관계는 알 수 없다.
- 힙은 완전이진트리구조이므로 배열 표현하여도 메모리 낭비가 없다.

(1) 최대힙에서 삽입 - O($\log_2 n$)

먼저, 최대힙에 새로운 노드를 삽입할 때 삽입될 위치는 다음과 같다.
이유는 힙은 기본구조가 **완전이진트리**를 만족해야 하기 때문이다.

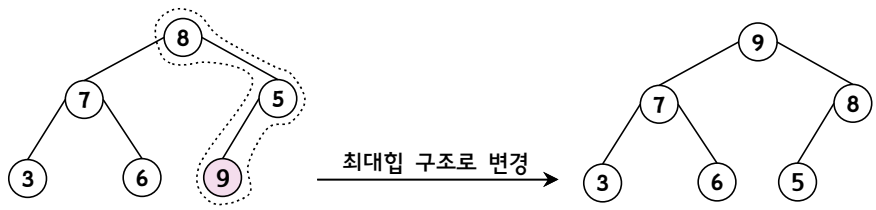


① 원소 2를 가지는 노드 삽입



• 새로운 노드 2가 삽입되었지만, 최대힙 조건을 **만족**하므로 추가적인 작업이 필요없다.

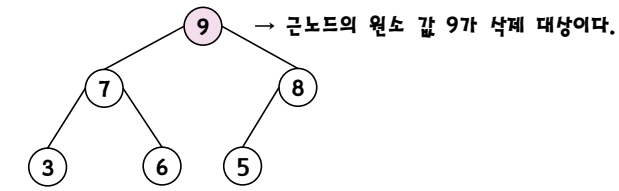
② 원소 9를 가지는 노드 삽입



• 새로운 원소 9를 가지는 노드가 삽입되면 최대힙 조건을 만족하지 않게 된다.
• 이러한 경우는 최대힙이 되도록 자식과 부모노드의 원소 값을 차례로 비교하여 교환한다.
• 새로 생성된 자식노드에 부모노드 값 5가 들어가고, 9는 부모노드에 임시로 들어가게 되고,
• 9는 또 다시 부모노드의 값 8과 비교하게 되고, 9와 8은 서로 위치가 교환된다.

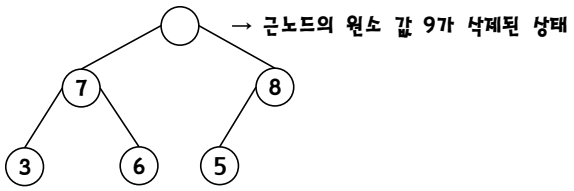
(2) 최대힙에서 삭제 - O($\log_2 n$)

먼저, 최대힙에서 **원소 삭제는 근노드**에서 실시한다.

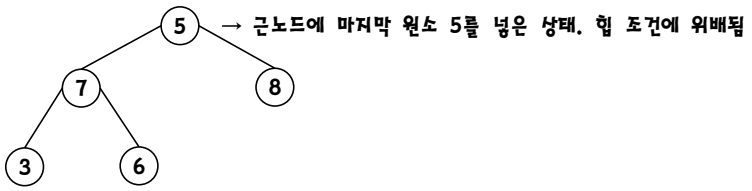


다음은 최대힙에서 삭제하는 과정이다.

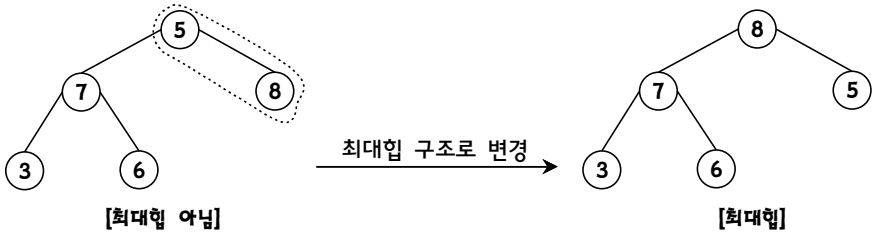
① 근노드의 원소 9를 삭제한다.



② 마지막 노드의 원소 5를 근노드에 넣는다. 마지막 노드는 없어진다.



③ 최대힙이 되도록 부모노드와 좌우 자식노드의 값을 비교하여 재구성한다.



(3) 최대힙의 삽입 알고리즘

다음은 배열을 이용한 최대힙의 삽입 알고리즘이다.

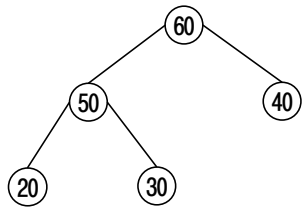
```
// 배열요소 0번째를 사용하지 않는 경우
#define SIZE 50
struct insa{
   int key;
   char irum[10];
}heap[SIZE];
int n = 0;

void insert_max_heap(struct insa data){
    int i;
    if(n == SIZE - 1){
      printf("The heap is full...!\n");
      exit(1);
    }
    i = ++n;
    while( (i != 1) && (data.key > heap[i/2].key) ){ //루트 위치가 아니면 키 비교
      heap[i] = heap[i/2]; //부모노드 값을 자식노드 위치에 대입
      i /= 2; //부모노드 위치를 구함(1/2씩 줄어들므로 log2 n)
    }
    heap[i] = data; //입력 키 값이 부모노드 값보다 작으면 인덱스 i에 저장됨
}
void main(){
    struct insa node;
    int i, j = 0;
    do{
      printf("\nKey...  "); scanf("%d", &node.key); if(node.key==-1) exit(1);
      printf("\nirum..  "); scanf("%s", node.irum);

      insert_max_heap(node);

      printf("\n Output of Heap\n");
      for(i=1; i<=n; i++) printf("%4d  %10s\n",heap[i].key, heap[i].irum);
    } while(j++<=SIZE-1);
}
```

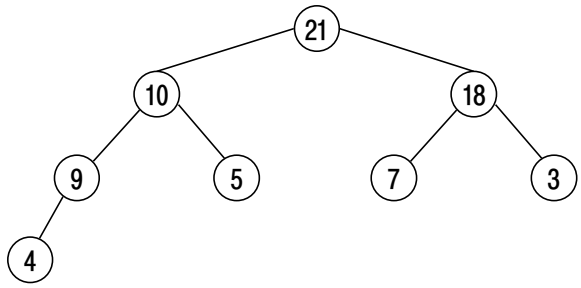
key	irum
40	순이
30	철수
50	연재
20	황소
60	한교



- 위에 제시된 트리는 5개의 자료를 입력한 결과의 최대힙 구조를 나타낸다.
- 최대힙 삽입 알고리즘은 삽입정렬 알고리즘과 비슷한 측면이 있다.

기출문제 분석

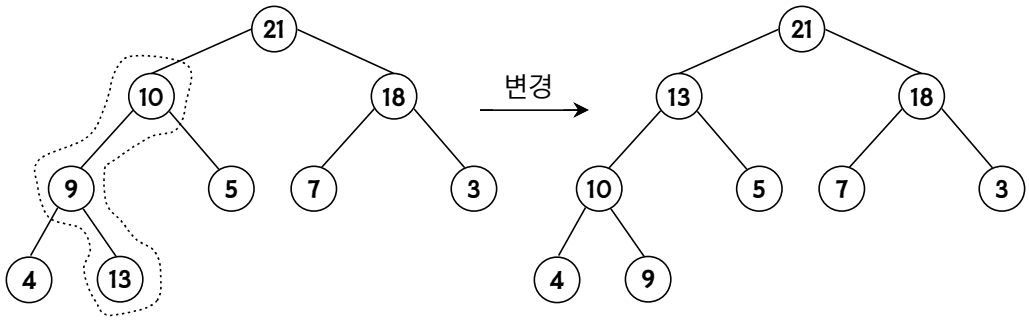
1. 다음 최대힙(max heap)에 13을 삽입하는 연산을 수행한 후에 루트 노드를 삭제하는 연산을 수행하였다. 그 결과, 루트 노드의 오른쪽 자식노드의 값으로 옳은 것은? (단, 삽입 및 삭제 연산은 최대 힙으로의 복원 과정을 포함한다) [2016년 국가 7급]



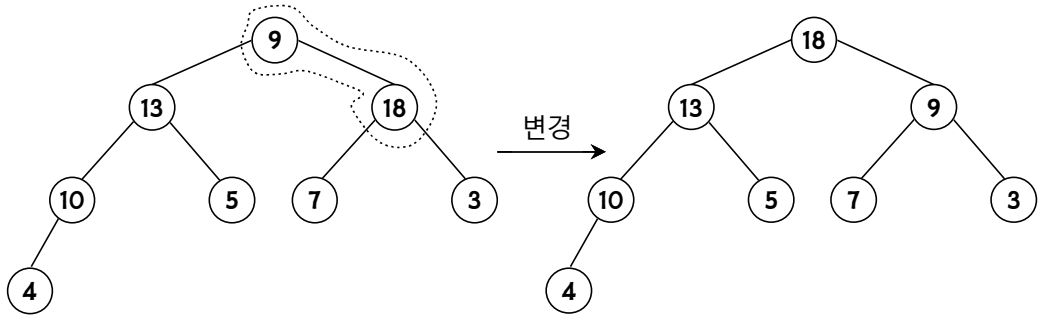
① 9 ② 10 ③ 13 ④ 18

♤ 최대힙에 삽입 / 삭제

• 13을 삽입



• 최대힙 삭제 – 최대힙에서 **원소 삭제는 근노드**에서 실시(마지막 노드 9를 근노드로)



정답 : ①

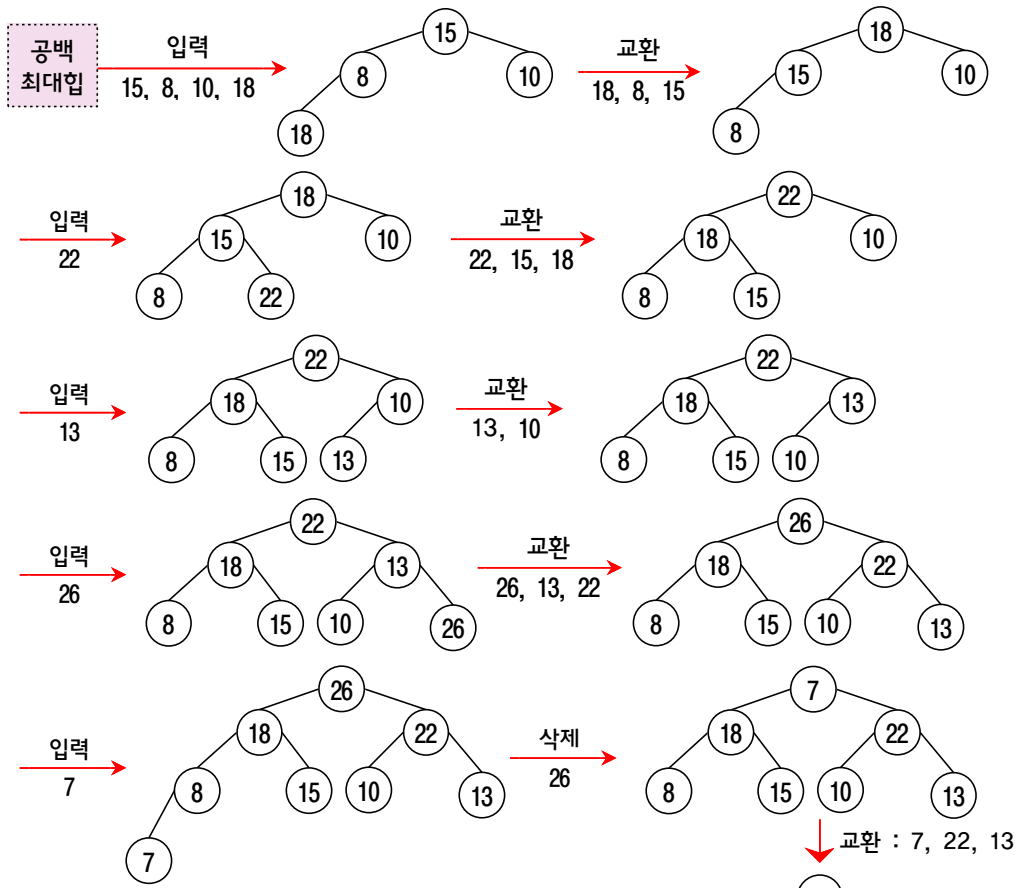
2. 루트노드의 인덱스를 1로 하는 1차원 배열을 이용하여 최대힙(max heap)을 구현한 후, 다음 데이터를 차례대로 하나씩 힙에 삽입하였다. 이후 힙 삭제 연산을 1회 수행한 후 배열의 인덱스 6에 저장된 데이터는? [2022년 국가 7급]

15, 8, 10, 18, 22, 13, 26, 7

---

① 8 ② 10 ③ 13 ④ 15

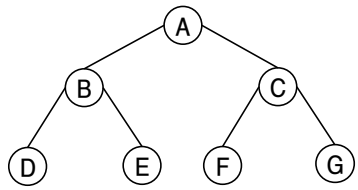
♤ 최대힙 – 15, 8, 10, 18, 22, 13, 26, 7



인덱스	1	2	3	4	5	6	7
데이터	22	18	13	8	15	**10**	7

정답 : ②

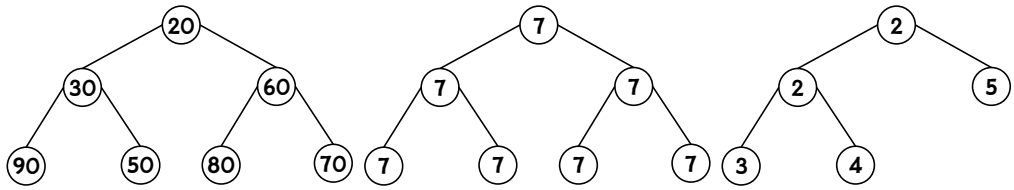
3. 다음 최소힙(min heap)에 대한 설명으로 옳지 않은 것은? [2020년 국가 7급]



① 완전이진트리(complete binary tree)이다.
② 노드 A는 가장 작은 키 값을 가져야 한다.
③ 노드 B와 노드 C가 같은 키 값을 가질 수 있다.
④ 노드 C는 노드 D보다 작은 키 값을 가져야 한다.

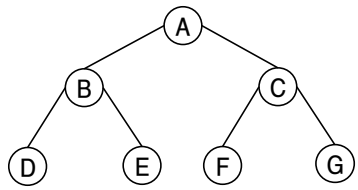
♤ **최소힙**

// 다음 3개의 트리는 모두 **최소힙**이다.(부모노드 값≤자식노드 값)



[최소힙 1] **[최소힙 2]** **[최소힙 3]**

- **최소힙**은 완전이진트리이면서, 부모노드 값≤자식노드 값이다.
- **최소힙**은 부모노드 값이 가장 작다.



↓

- 노드 C는 노드 D보다 작은 키 값을 가져야 한다.(×)

↓

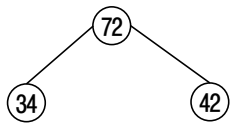
↓ 올바르게 고치면

↓

- 노드 C와 노드 D가 가지는 값은 **최소힙** 조건에 무관하다.

정답 : ④

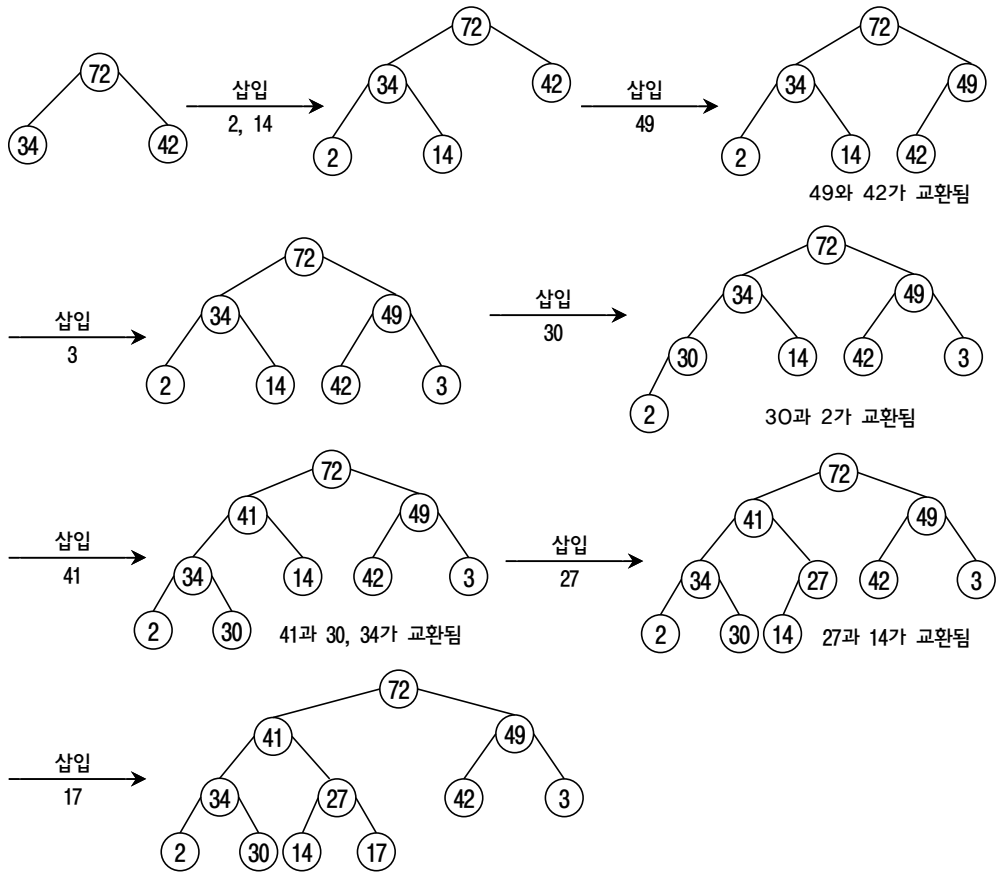
4. 다음에 주어진 최대힙(max heap)에 키 값이 2, 14, 49, 3, 30, 41, 27, 17인 8개의 데이터를 순서대로 삽입하여 최대힙을 구성하였다. 구성된 최대힙에서 모든 리프(leaf) 노드의 키 값의 합은? [2020년 국가 7급]



① 63 ② 73 ③ 108 ④ 13

♤ **최대힙에 삽입**

- 최대힙 : 부모노드 값≥자식노드 값 이면서, 완전이진트리이다.
- 추가 입력 : **2, 14, 49, 3, 30, 41, 27, 17**

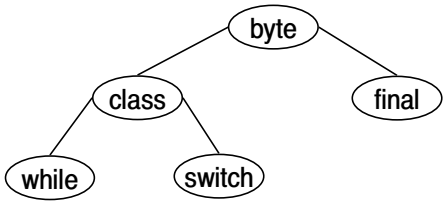


- 리프(leaf) 노드의 키 값 합 = 2 + 30 + 14 + 17 + 42 + 3 = 108

정답 : ③

5. 다음은 완성된 최소힙 트리이다. 입력 순서로 옳은 것은? [2020년 군무 7급]

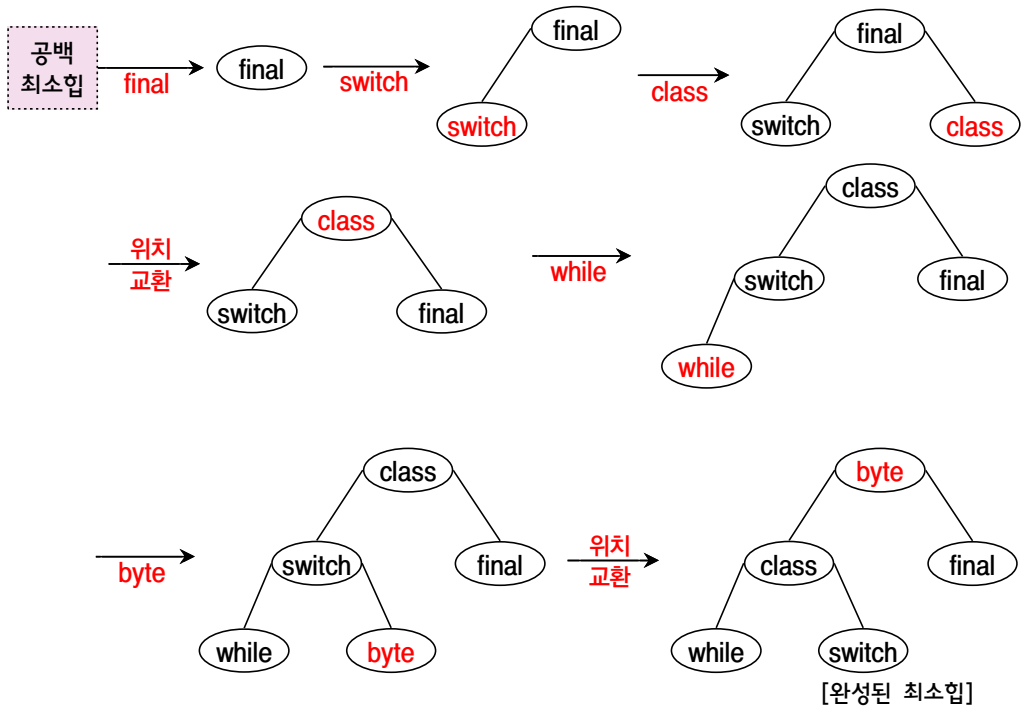
---<최소힙 트리>----------



---

① 입력 순서 : byte, final, switch, class, while
② 입력 순서 : final, switch, class, while, byte
③ 입력 순서 : switch, class, while, byte, final
④ 입력 순서 : class, while, byte, final, switch

♤ 최소힙 – 입력 순서 : final, switch, class, while, byte



• 이런 유형의 문제는 주어진 입력 순서를 이용하여 힙 트리를 그려보면 된다.(시간이 필요)

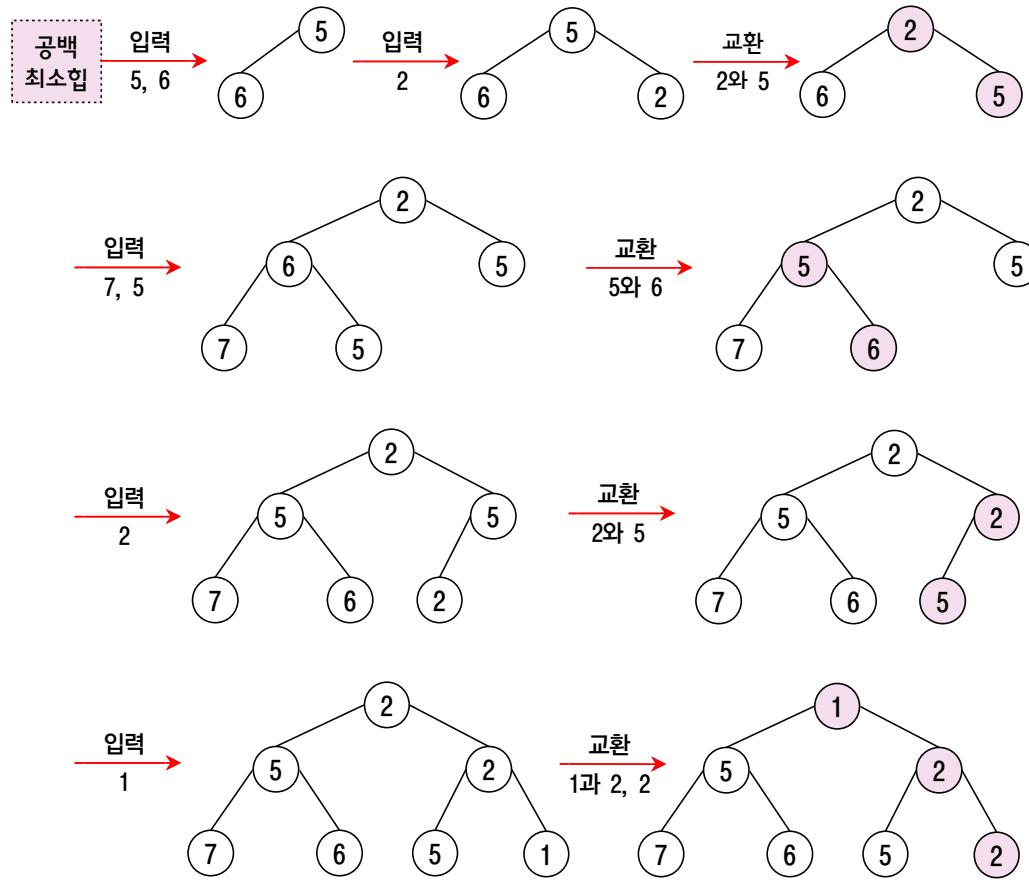
정답 : ②

6. 다음과 같은 입력 데이터를 차례로 삽입하면서 최소힙 트리를 구성하였다. 완성된 최소힙 트리에 대한 설명으로 옳지 않은 것은? [2020년 군무 7급]

입력 데이터 : 5, 6, 2, 7, 5, 2, 1
---

① 트리의 높이는 3이다.(단, 루트노드의 높이는 1로 한다)
② 트리의 중위순회 결과는 1, 2, 2, 5, 5, 6, 7이다.
③ 트리의 모든 단말노드는 같은 레벨에 있다.
④ 루트노드의 좌측 자식노드 값은 5이다.

♤ **최소힙**



• 트리의 중위순회(좌, 중, 우) 결과 : 7, 5, 6, 1, 5, 2, 2

정답 : ②

7. 비어 있는 최소힙(min heap)에 <보기>에 주어진 데이터를 순서대로 하나씩 삽입 연산을 수행하여 모두 삽입한 후, 2번의 삭제 연산을 수행하였다. 최소힙의 가장 마지막 레벨의 가장 왼쪽 노드에 저장된 데이터는? (단, 최소힙은 완전이진트리로 구현한다) [2022년 서울 7급]

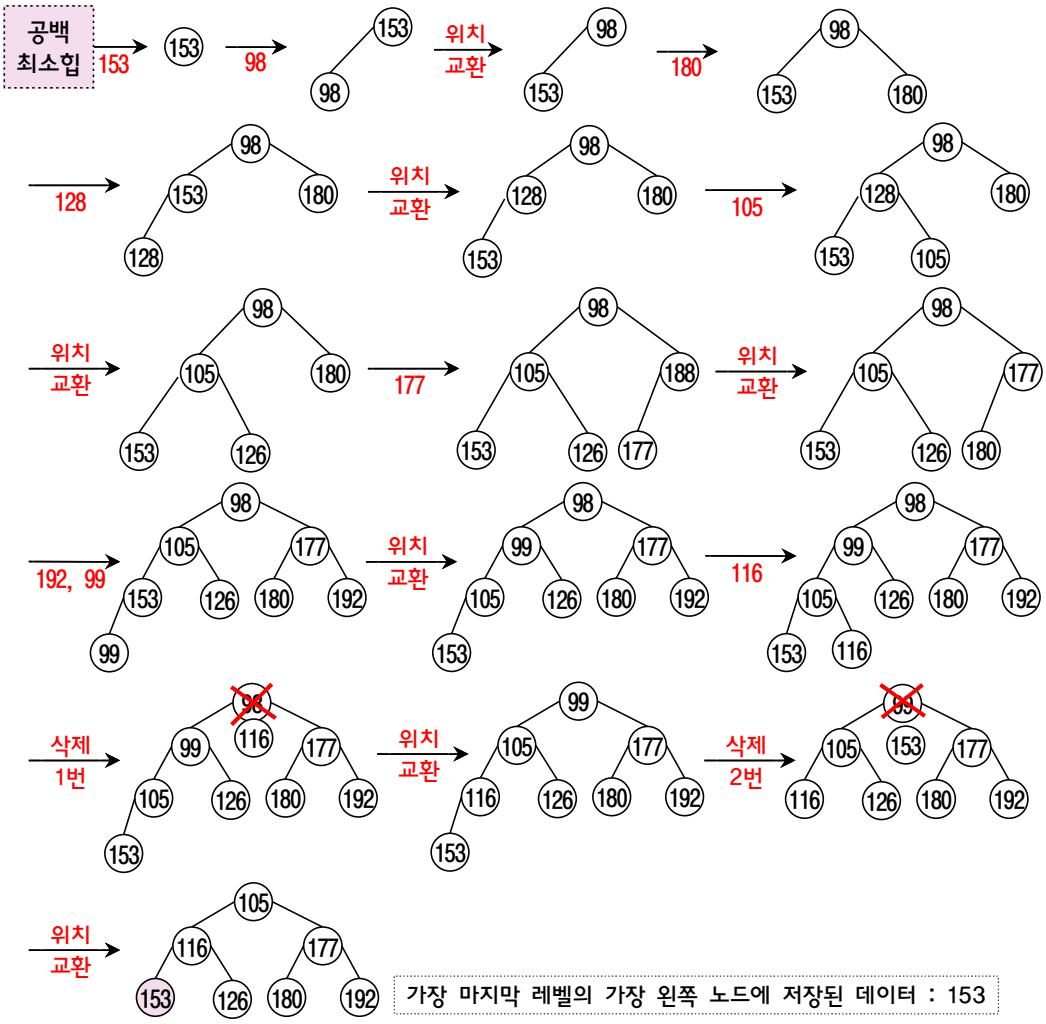
-----<보기>----------

153, 98, 180, 128, 105, 177, 192, 99, 116

---

① 128 ② 153 ③ 177 ④ 192

♤ 최소힙



정답 : ②

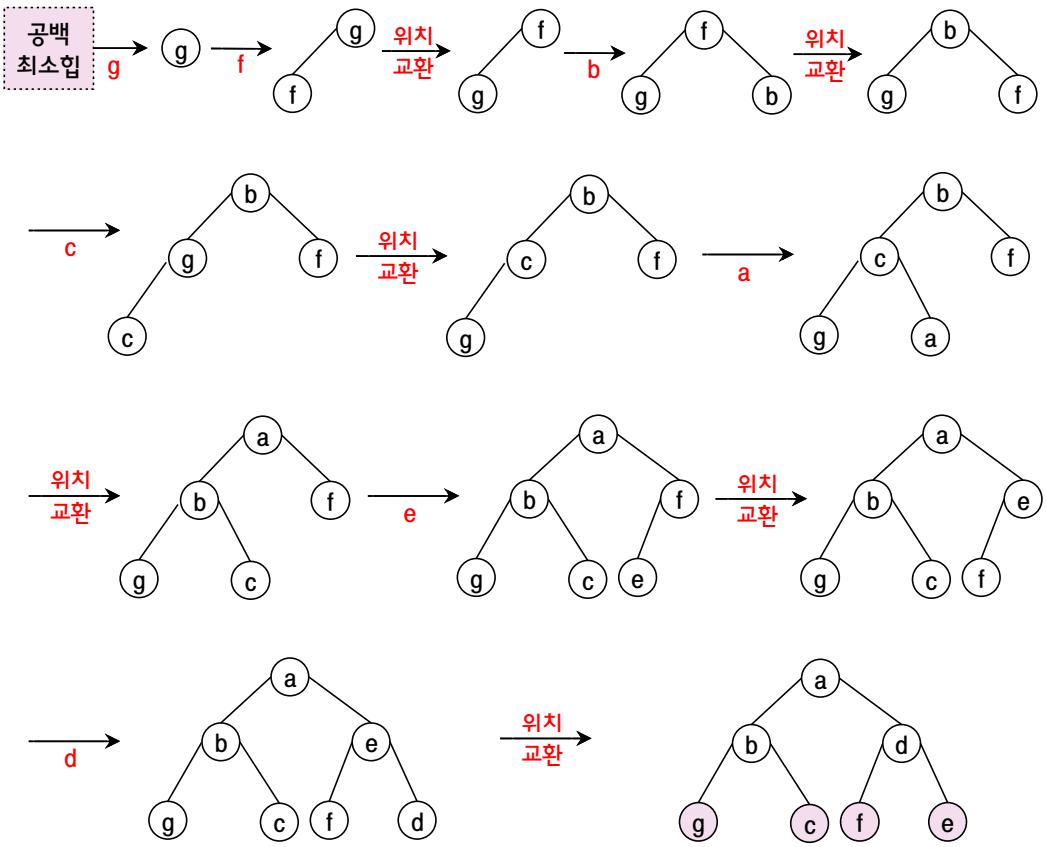
8. 다음 문자열 K의 각 문자를 최소힙(min heap)의 자료구조에 차례로 삽입하였을 때, 단말노드(leaf node)들에 저장된 문자들을 왼쪽부터 차례로 나열한 것으로 옳은 것은? (왼쪽→오른쪽) [2022년 군무원 7급]

삽입할 문자열(K) : g f b c a e d

---

① g c f e　　② g f e d
③ d g f e　　④ f g e d

♤ 최소힙

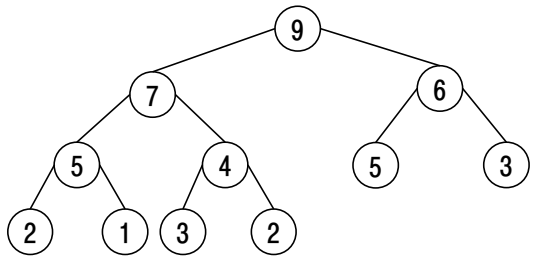


· 단말노드 왼쪽→오른쪽 : g c f e

정답 : ①

9. <보기>의 최대힙(max heap)에 8을 삽입한 후, 최댓값을 삭제하는 연산을 수행하였다. 이때, 루트노드의 좌 · 우 자식노드의 값은? [2020년 서울 7급]

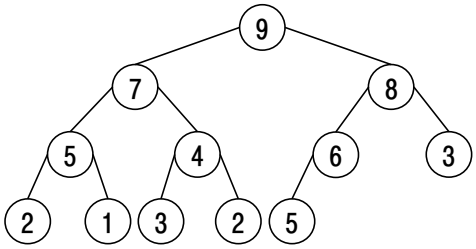
----<보기>----------



	좌	우
①	7	5
②	7	6
③	7	8
④	8	7

♤ **최대힙**

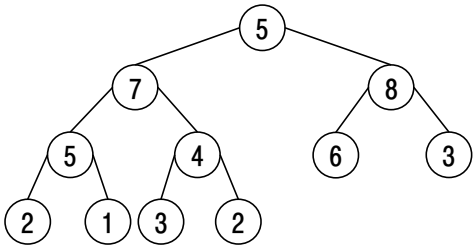
삽입 : 8



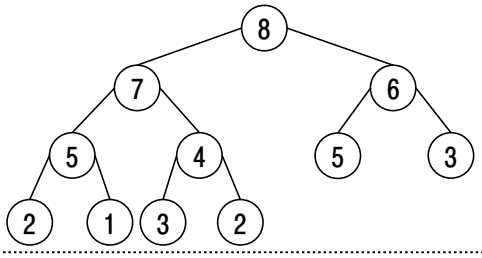
• 8이 삽입된 후 : 8, 5, 6의 위치가 교환됨

삭제 : 9

• 루트 9를 삭제하려면
• 9를 삭제하고, 마지막 노드 5를 루트로 이동



• 9가 삭제된 모습
• 마지막 노드 5가 루트로 이동됨

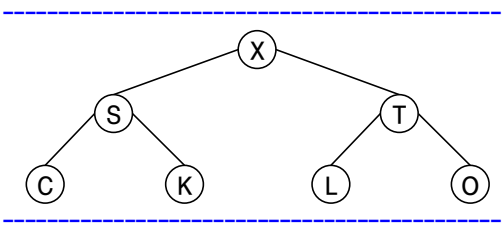


• 그리고, 부모가 큰 값이 되도록 위치 교환
• 5, 8, 6이 교환됨

• 루트노드의 좌 · 우 자식노드의 값 : **7, 6**

정답 : ②

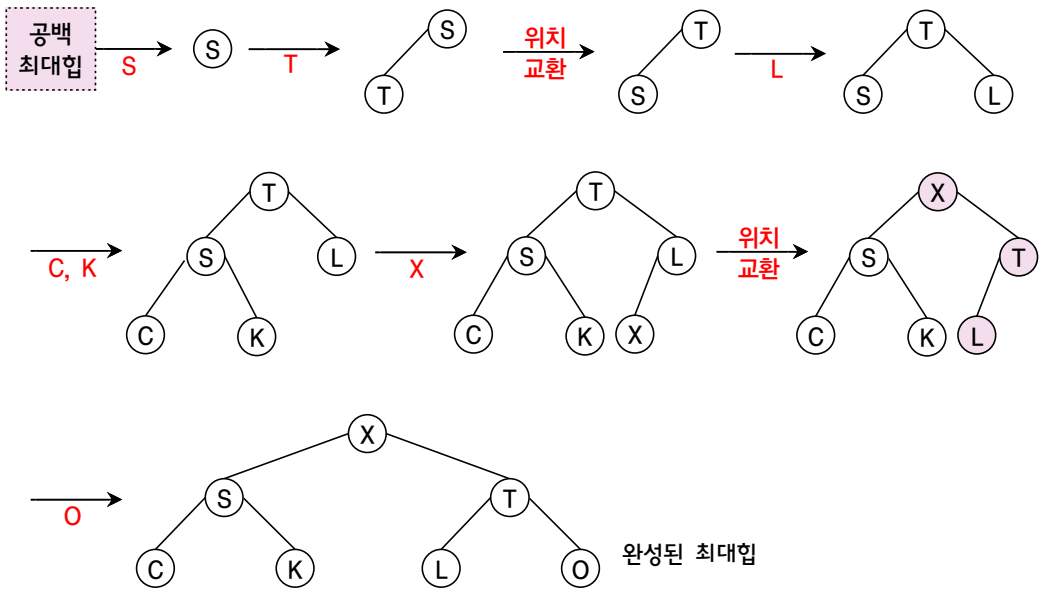
10. 아래 그림의 heap은 부모 node의 값이 자식 node들보다 크거나 같은 max heap이다. 이와 같은 모양의 heap을 만들어 낼 수 있는 삽입 순서로 가장 옳은 것은? [2022년 군무원 7급]



① C, T, S, X, K, L, O
② S, L, O, X, T, C, K
③ C, K, O, L, T, S, X
④ S, T, L, C, K, X, O

♤ 최대힙(max heap)

④ S, T, L, C, K, X, O에 대한 최대힙 생성 과정



• 이런 유형의 문제는 주어진 입력 순서를 이용하여 힙 트리를 그려보면 된다.(시간이 필요)

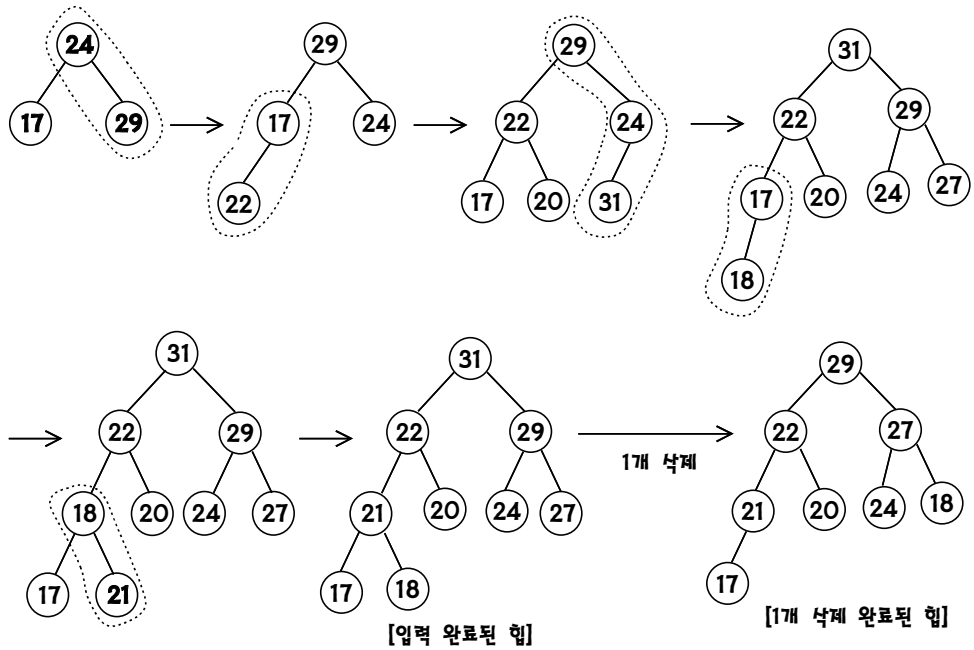
정답 : ④

11. 최대힙으로 우선순위큐를 구현하려 한다. 우선순위를 나타내는 9개의 데이터가 큐에 다음과 같은 순서대로 삽입되었다. 1개의 데이터가 큐에서 삭제된 후, 재 정렬된 힙에서 가장 마지막 원소는 무엇인가?(단, 숫자가 클수록 우선순위가 높다고 가정한다) [2007년 국가 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.

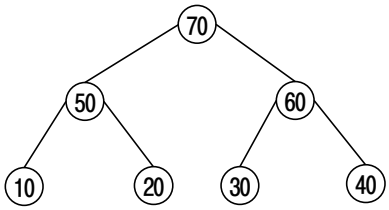
정답 : ①

◈ **힙에서 자료 삽입/삭제**

- 이 문제는 처음 발표된 '정답가안'에서 정답이 '② 18'로 발표되었다.
  해서, 이의 신청을 하였고, 최종 답안은 '① 17'로 변경되었다.
- 2003년도 기술고시 '자료구조론' 시험 문제에서도 이런 유형의 문제에 대한 이의 신청이 많았다.
  최종 정답은 위에서 설명한 원리대로 적용한 것만 인정되었다.
- 힙구조 운영과 힙정렬에서 힙구조를 이용하는 원리는 다르다.

힙구조	자료가 입출력되는 시점에 힙 조건에서 벗어나면 힙구조로 변경시킨다.
힙정렬	정렬 대상 자료를 트리에 모두 입력한 후 힙구조로 변경시킨다.

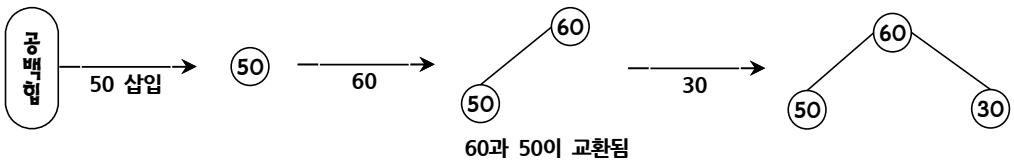
12. 그림과 같은 최대힙(max heap)을 구성하기 위한 삽입 순서로 옳은 것은? [2017년 국가 7급]

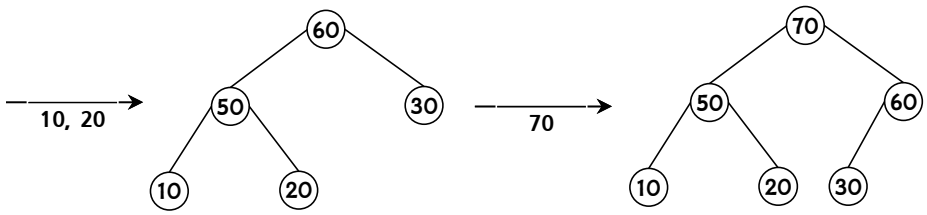


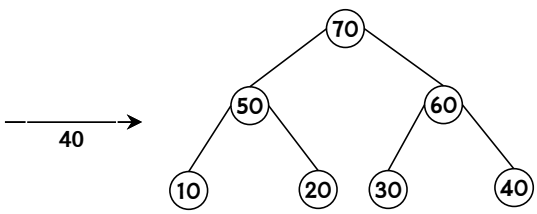
① 50, 30, 40, 70, 60, 10, 20
② 10, 60, 50, 70, 20, 30, 40
③ 10, 20, 40, 30, 60, 50, 70
④ 50, 60, 30, 10, 20, 70, 40

♤ 최대힙

• 최대힙 : 부모노드 값≥자식노드 값 이면서, 완전이진트리이다.
• 입력 : 50, 60, 30, 10, 20, 70, 40





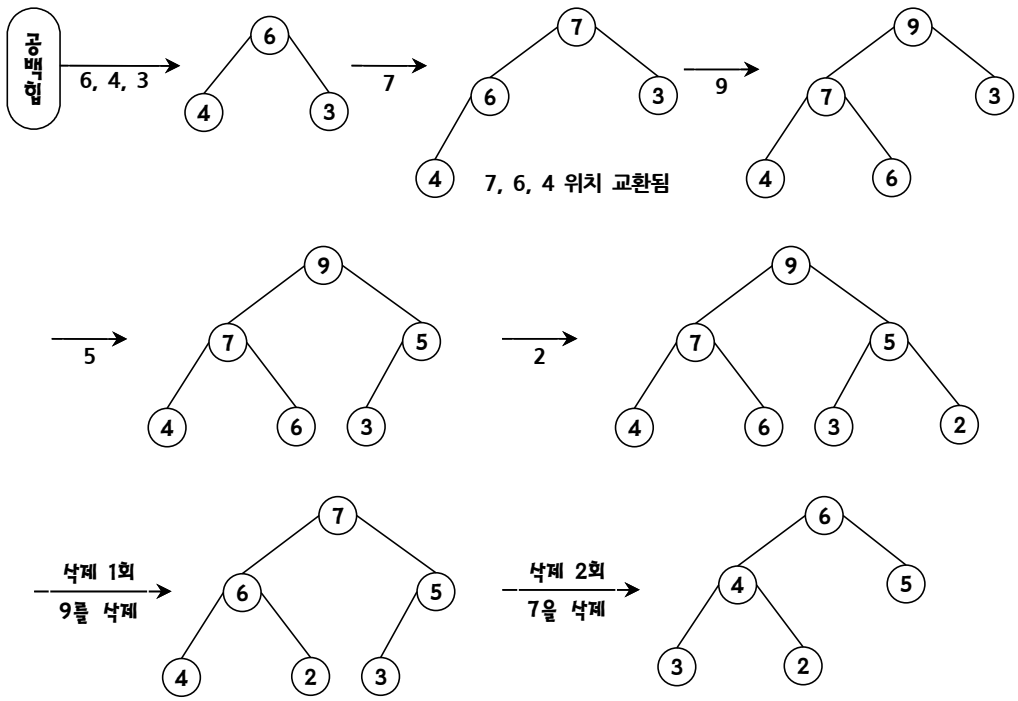


정답 : ④

13. 숫자 6, 4, 3, 7, 9, 5, 2를 차례로 삽입하여 최대힙을 구성한 후에 삭제 연산을 2회 수행하였다면, 최대힙을 구성하는 배열의 내용으로 알맞은 것은? [2003년 기술고시]

	[1]	[2]	[3]	[4]	[5]
①	6	5	4	3	2
②	6	5	4	2	3
③	6	5	3	4	2
④	6	4	5	2	3
⑤	6	4	5	3	2

◈ 최대힙 – 자료가 삽입/삭제되면서 최대힙이 변해가는 과정이다.



• 삭제 연산 2회한 결과의 최대힙을 근노드부터 레벨 운행하면 : **6, 4, 5, 3, 2**

정답 : ⑤

☞ 이 유형의 문제는 과거에 출제될 때마다 거의 **이의 신청**이 있었고,
정답이 잘못되어 **수정**된 경우도 있었다.
아무튼, 힙에서 삽입/삭제는 위에서 설명한 것처럼 처리하면 된다.

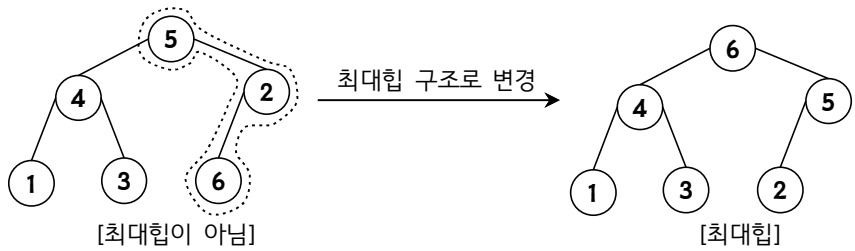
14. 다음은 빈 상태의 힙(heap) 배열에 1~8까지의 키 순서로 삽입이 이루어질 때, 힙이 형성되는 과정을 순서대로 나타난 그림이다. 빈칸들에 알맞은 것은? [2010년 국가 7급]

```
1
2  1
3  1  2
4  3  2  1
5  4  2  1  3
(           )
7  4  6  1  3  2  5
(                 )
```

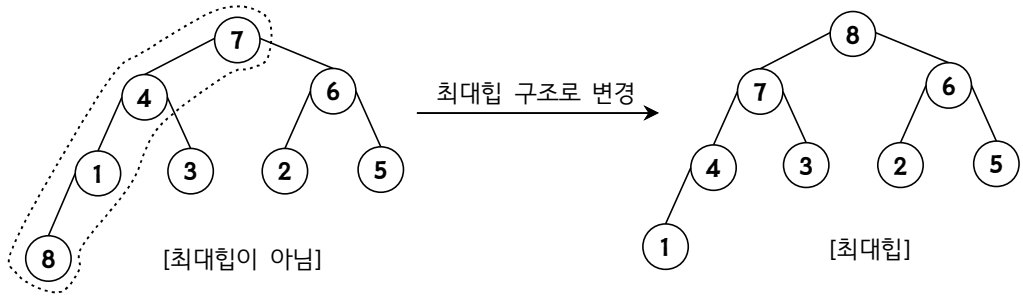
① (6 4 1 5 3 2), (8 7 6 4 3 2 5 1)
② (6 4 5 1 2 3), (8 7 6 4 3 5 2 1)
③ (6 4 5 1 3 2), (8 7 6 3 4 2 5 1)
④ (6 4 5 1 3 2), (8 7 6 4 3 2 5 1)

♤ **최대힙 – 완전이진트리, 부모노드 값 ≥ 자식노드 값**

- 1에서 5까지 입력되는 과정은 생략
- 6이 입력되고, 힙구조 변경하면 – (6 4 5 1 3 2)



- 7이 입력되는 과정은 생략
- 8이 입력되고, 힙구조 변경하면 – (8 7 6 4 3 2 5 1)



정답 : ④

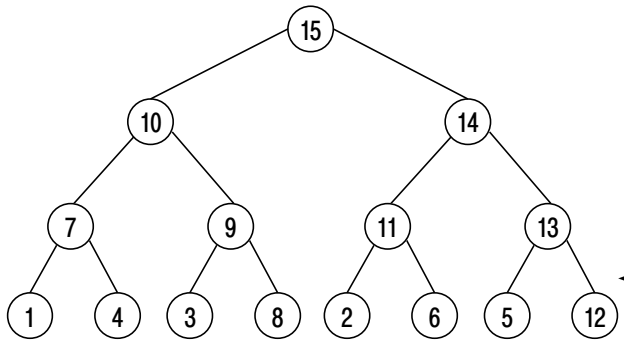
15. 공백인 최대힙에 정수 1부터 15까지가 임의 순으로 입력될 때, 옳지 않는 설명은? [2019년 군무 7급]

① 15는 항상 근노드이다.
② 12는 단말노드에 있을 수 없다.
③ 13은 단말노드에 있지 않다.
④ 2는 항상 단말노드이다.

♤ **공백인 최대힙에 정수 1부터 15까지 임의 순으로 입력**

- 최대힙은 완전이진트리이다.
- 최대힙 : 부모노드 값 ≥ 자식노드 값(단, 자식이 있을 때)

- 이 문제는 정수 1부터 15까지 임의 순으로 입력되므로 모든 경우의 수를 고려해야 한다.
- 다음은 공백인 최대힙에 정수 1부터 15까지 차례로 입력되었을 때, 최종 모습이다.



- 15는 가장 큰 값이므로 최대힙 특성에 의해 항상 근노드가 된다.

- 12는 입력 순서에 따라 단말노드에 있을 수 있다.(주어진 그림처럼)
- 최종 힙구조의 레벨이 4이므로 12는 단말노드에 위치할 수 있다.

- 13은 단말노드에 있지 않다.
- 최종 힙구조의 레벨이 4이므로 13은 단말노드에 위치할 수 없다.

- 2는 항상 단말노드이다. 2는 일찍 입력되어도 작은 값이므로 단말노드에 위치하게 된다.

정답 : ②

16. 다음은 최대힙을 배열로 표현한 것이다. 옳지 않는 설명은? [2019년 군무 7급]

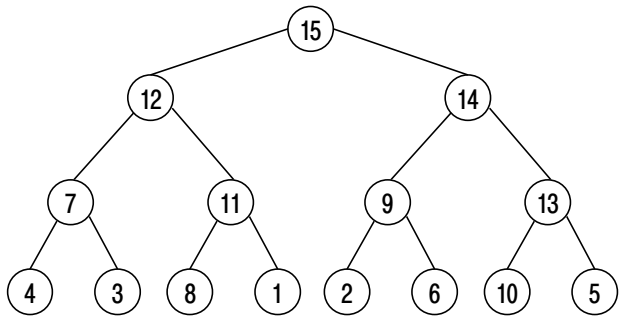
15	12	14	7	11	9	13	4	3	8	1	2	6	10	5
a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]

① 1회 삭제후 13은 a[3]에 저장되었다.
② 2회 삭제후 10은 a[3]에 저장되었다.
③ 3회 삭제후 8은 a[5]에 저장되었다.
④ 4회 삭제후 6은 a[6]에 저장되었다.

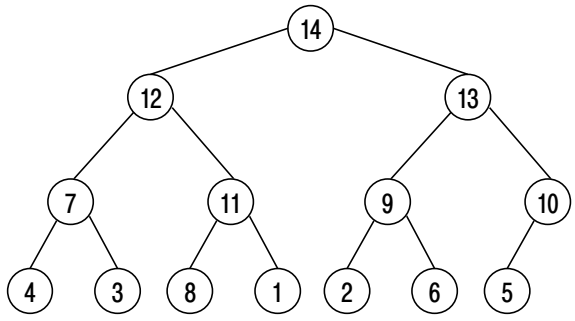
♤ **최대힙**

15	12	14	7	11	9	13	4	3	8	1	2	6	10	5
a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]

↓ 힙트리로 표현하면



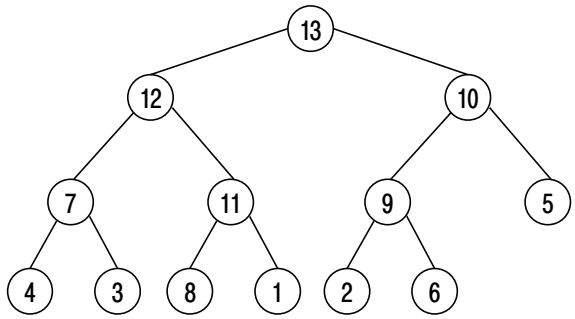
↓ 최대힙에서 삭제 대상은 루트이다.
↓ 삭제 1회(루트 15 삭제)



↓ 삭제 1회(루트 15 삭제) 후에 배열구조

14	12	13	7	11	9	10	4	3	8	1	2	6	5	
a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]

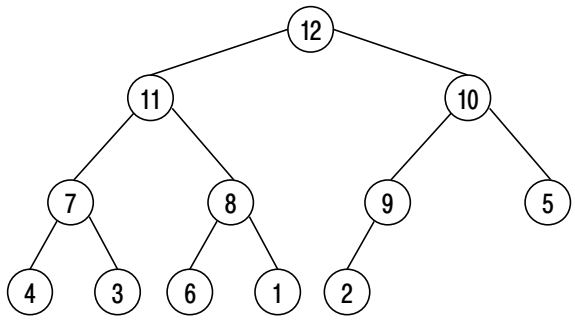
↓ 삭제 2회(루트 14 삭제)



↓삭제 2회(루트 14 삭제) 후에 배열구조

a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]
13	12	10	7	11	9	5	4	3	8	1	2	6		

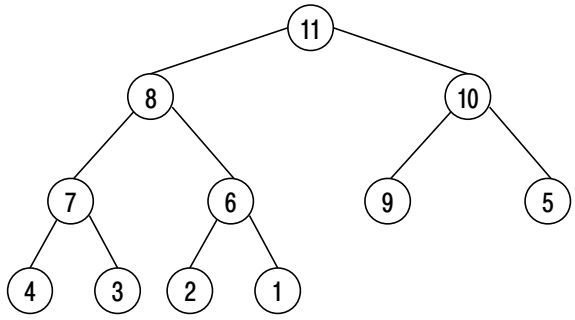
↓삭제 3회(루트 13 삭제)



↓삭제 3회(루트 13 삭제) 후에 배열구조

a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]
12	11	10	7	8	9	5	4	3	6	1	2			

↓삭제 4회(루트 12 삭제)



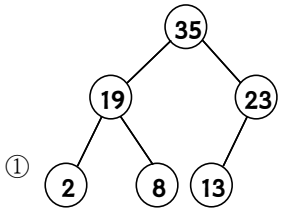
↓삭제 4회(루트 13 삭제) 후에 배열구조

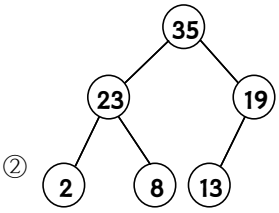
a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]
11	8	10	7	6	9	5	4	3	2	1				

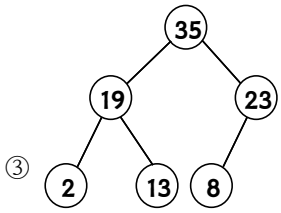
정답 : ④

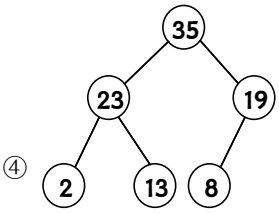
17. 다음 데이터들을 공백힙(empty heap)에 차례대로 삽입하여 최대힙(max heap)을 생성하였다. 생성된 최대힙에 삭제 연산을 두 번 수행한 후의 결과는? [2012년 국가 7급]

2, 13, 55, 8, 35, 47, 23, 19

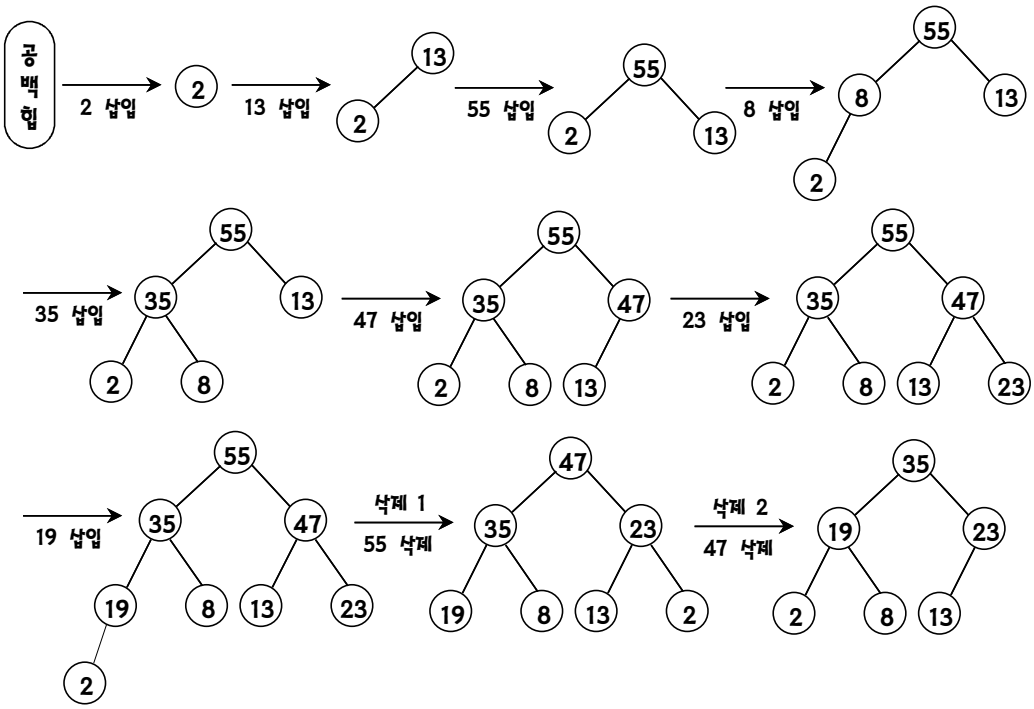
① 

② 

③ 

④ 

♤ 최대힙 – 힙에서 삭제는 기본적으로 근노드이다.



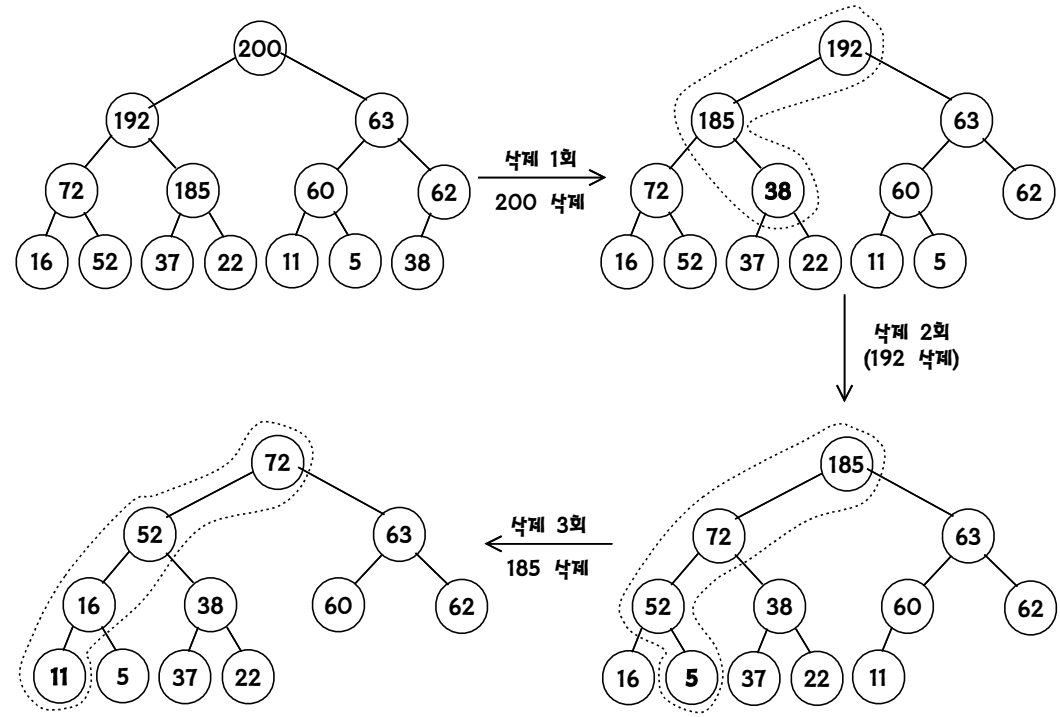
정답 : ①

18. 다음과 같이 14개의 정수가 최대힙(max heap)을 표현하는 배열의 1번 위치부터 14번 위치까지 저장되어 있다. 이 배열에서 최대값을 제거하는 연산을 3번 수행한 후, 최대힙의 1번 위치부터 11번 위치까지의 위치에 저장되어 있는 수들을 올바르게 나열한 것은?(단, 제거 연산 시 가급적 최대힙의 내용 변경을 최소화 한다고 가정한다) [2009년 국가 7급]

데이터 : 200 192 63 72 185 60 62 16 52 37 22 11 5 38

① 72 63 62 60 52 38 37 22 16 11 5
② 72 52 63 38 16 60 62 37 22 11 5
③ 72 52 63 16 38 60 62 11 5 37 22
④ 72 52 63 16 38 60 62 5 11 37 22

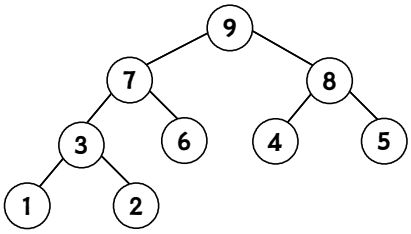
♤ 최대힙에서 삭제 연산



• 삭제 연산을 3회 실시한 후에 근노드부터 차례로 읽으면 된다.(레벨오더운행)

정답 : ③

19. 다음 최대힙(max heap) 트리를 일차원 배열에 표현할 경우, 배열에서 최댓값을 두 번 삭제한 후 남아있는 배열의 상태로 옳은 것은? (단, 최대힙 배열에서 첫 번째 요소는 비어있다) [2015년 국가 7급]



최대힙의 배열 표현 :

-	9	7	8	3	6	4	5	1	2

---

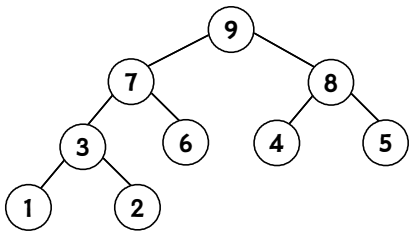
① | - | 7 | 3 | 6 | 4 | 5 | 1 | 2 | | |

② | - | 7 | 6 | 5 | 3 | 4 | 1 | 2 | | |

③ | - | 7 | 6 | 5 | 3 | 1 | 4 | 2 | | |

④ | - | 7 | 6 | 5 | 4 | 3 | 2 | 1 | | |

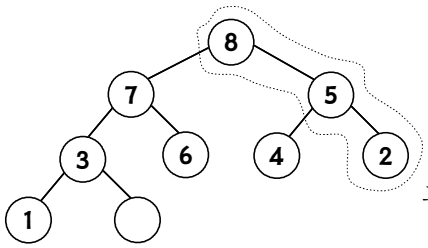
♤ 최대힙 – 두 번 삭제



↓

↓첫 번째 삭제, 근노드 값 9를 삭제하고 2를 근노드에 넣는다.

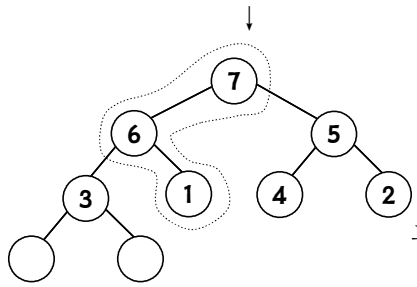
↓



부모노드≥자식노드이 되도록 교환한 결과

↓

↓두 번째 삭제, 근노드 값 8을 삭제하고 1을 근노드에 넣는다.



부모노드≥자식노드이 되도록 교환한 결과

• 삭제 연산 2회 실행한 결과의 힙을 근노드부터 레벨 운행하면

-	7	6	5	3	1	4	2		

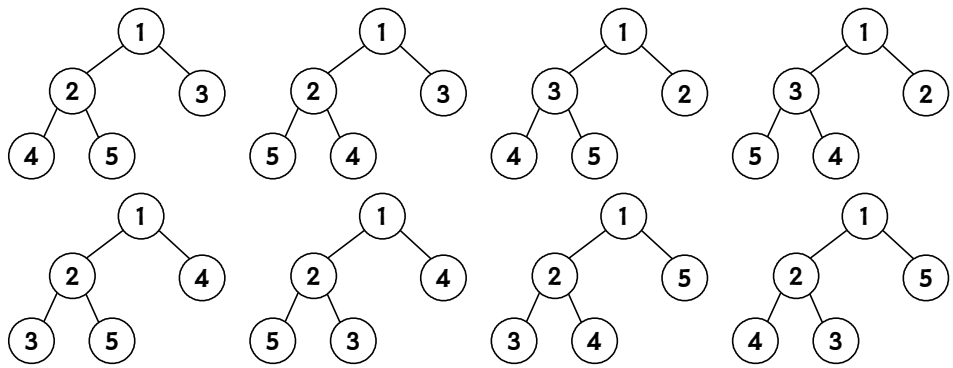
정답 : ③

20. 다음 다섯 개의 숫자를 갖는 최소힙(min-heap)을 만들려고 한다. 구성할 수 있는 서로 다른 최소 힙의 경우의 수는? [2014년 국가 7급]

1, 2, 3, 4, 5
---

① 1개 ② 6개
③ 8개 ④ 10개

♤ 최소힙의 경우의 수 – 8개



정답 : ③

21. 최대 히프(max heap) 구조의 특성으로 가장 옳지 않은 것은? [2018년 서울 7급]

① 부모노드(parent node)의 키 값이 자식노드(child node)의 키 값 보다 항상 크거나 같다.
② 왼쪽 서브트리(subtree)에 있는 노드들의 키 값들은 오른쪽 서브트리(subtree)에 있는 노드들의 키 값들보다 항상 작거나 같다.
③ 루트노드(root node)의 키 값이 최댓값이다.
④ 완전이진트리(complete binary tree)의 특성을 갖는다.

♤ **최대 히프**

- 왼쪽 서브트리(subtree)에 있는 노드들의 키 값들은 오른쪽 서브트리(subtree)에 있는 노드들의 키 값들보다 항상 작거나 같다.(×)
  → 최대 히프는 부모노드의 키 값이 자식노드의 키 값 보다 항상 크거나 같다.
  → 좌우 서브트리에 있는 노드들의 키 값들은 서로 상관이 없다.

정답 : ②