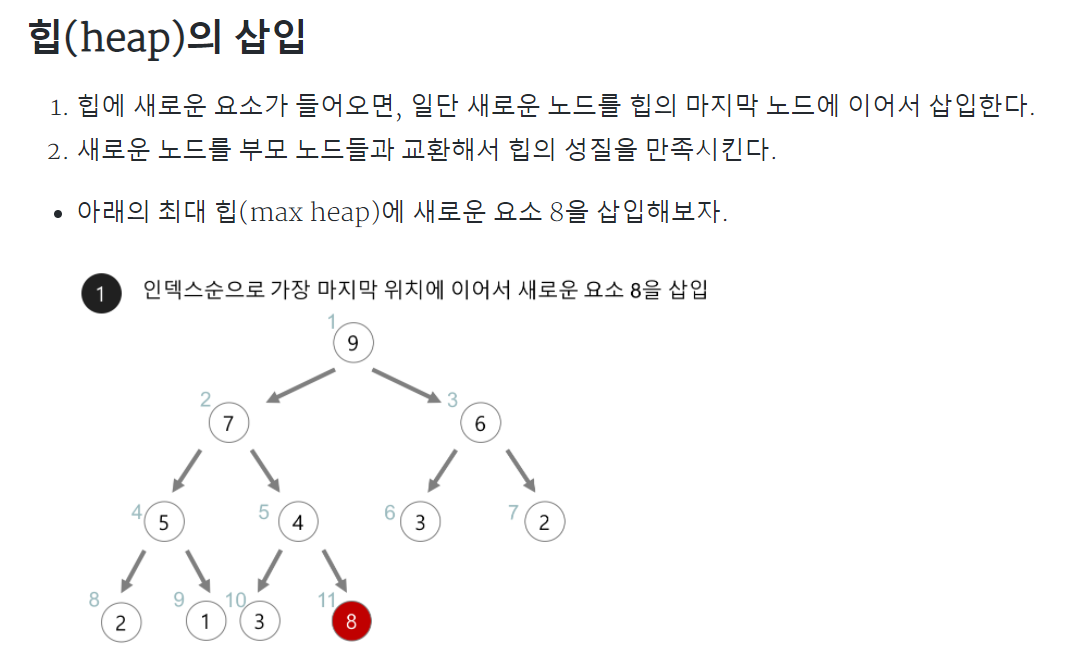
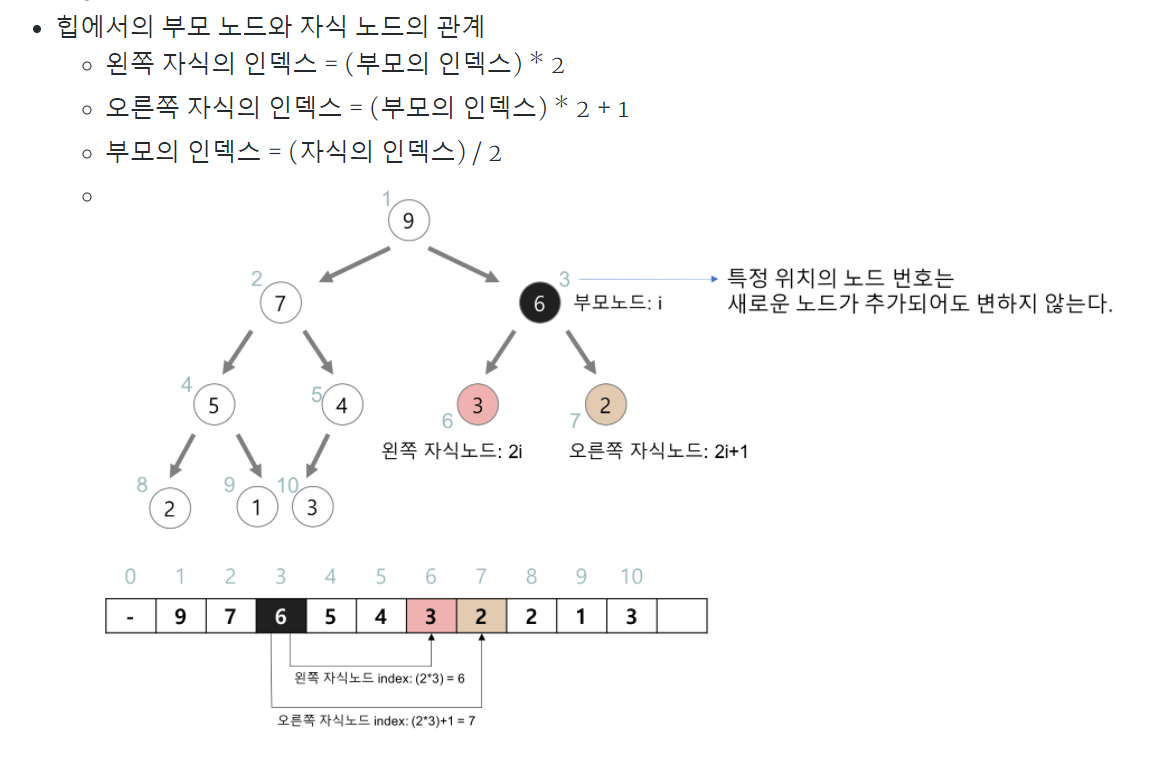
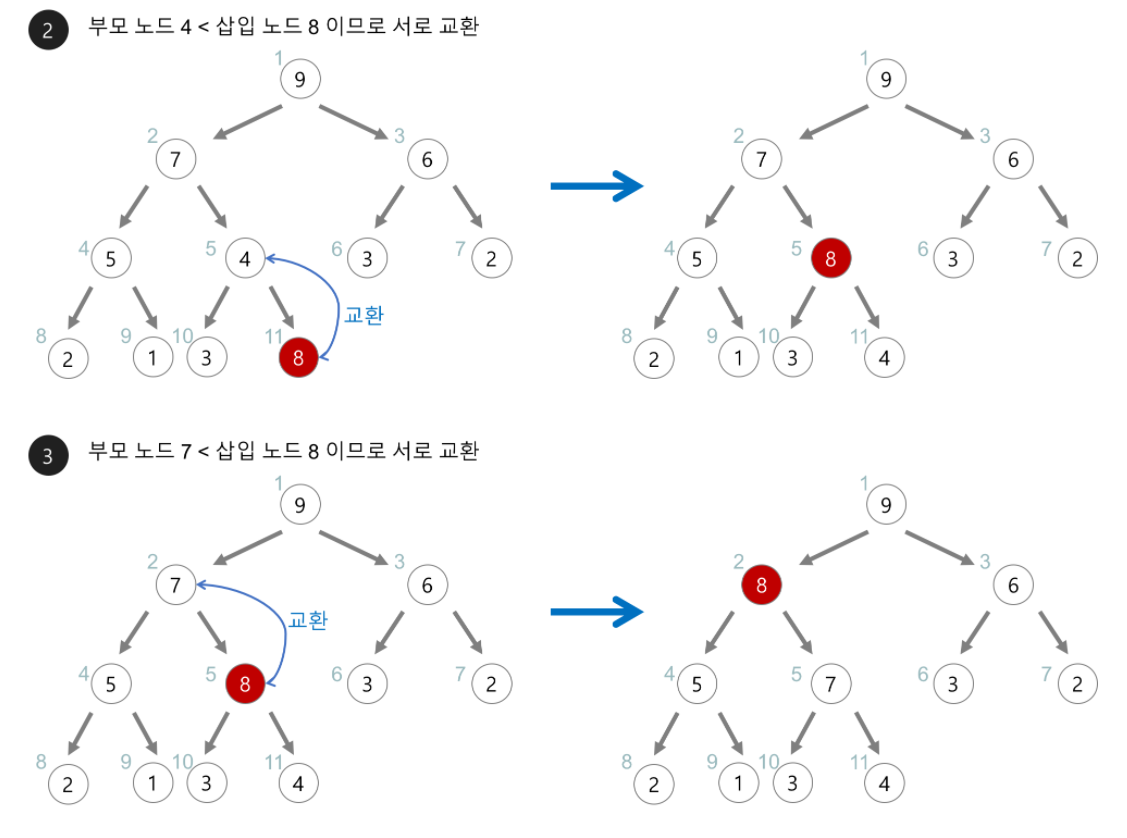
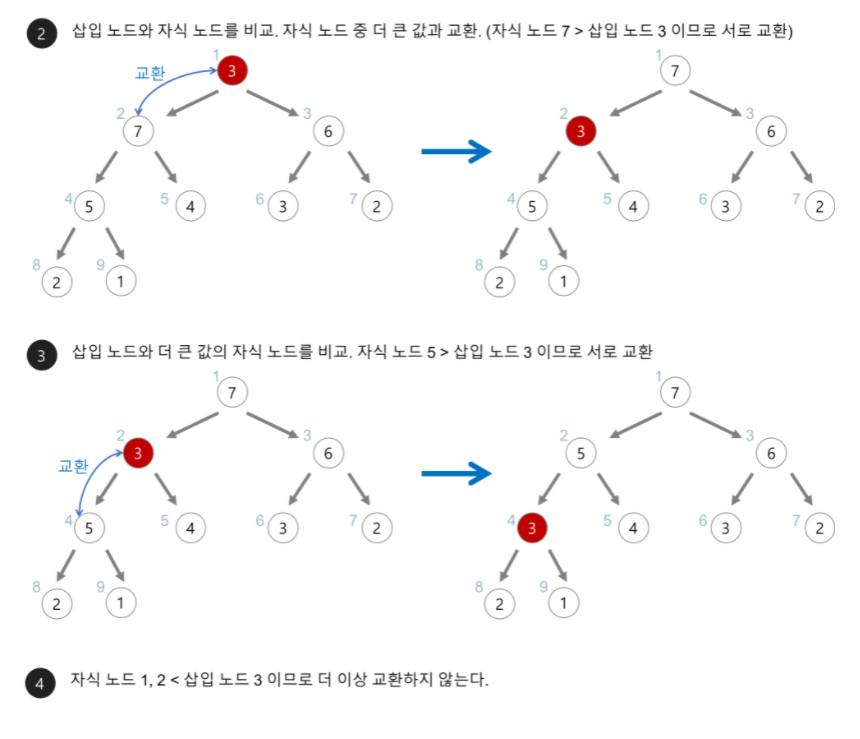
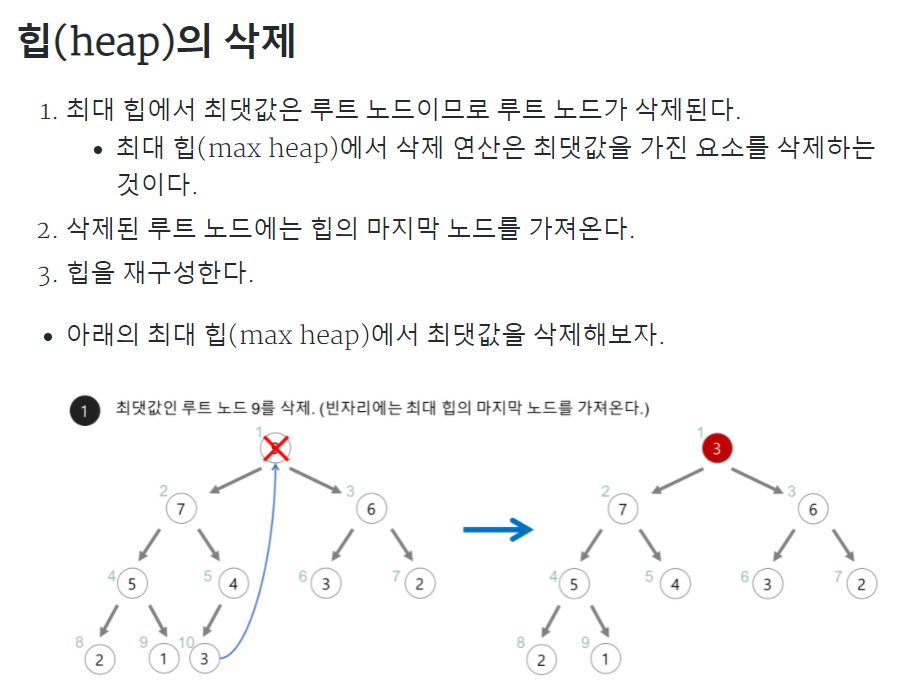
텍스트이(가) 표시된 사진

자동 생성된 설명







시간 복잡도

시간 복잡도는 데이터를 넣을 때도 O(lgN)이고 뺄 때도 O(lgN)이라 고른 성능을 보입니다. N개의 데이터를 모두 빼면 정렬이 되기 때문에 힙 정렬의 시간 복잡도는 O(NlgN)이 됩니다.

Python heapq를 사용하여 배열을 정리해 보자.

import heapq

def heapsort(a):

    h = []

    result = []

    # 모든 원소를 차례대로 힙에 삽입(heapq가 자동적으로 최소 힙을 만들어 줌.)

    for value in a:

        heapq.heappush(h, value)

    # 힙에 삽입된 모든 원소를 차례대로 꺼내어 담기.(heapq가 최솟값인 루트 노드 삭제 후 빈자리에 마지막 노드를 가져온 후 최소 이진 트리를 다시 만들어 줌.)

    for \_ in range(len(h)):

        result.append(heapq.heappop(h))

        print(result)

    return result

arr = [2, 3, 5, 6, 4]

res = heapsort(arr)

print(res)