1. 최우선순위 원소는 가장 큰 수이다.
2. 코드 참고
3. 4장에서 구현된 배열 Stack에서 가장 큰 수를 다루기 위해서는 Push와 Pop 둘 중 하나는 O(2N)이고, 나머지는 O(1)이다. Push 과정에서 O(1)로 값을 그냥 넣으면 Pop 과정에서 가장 큰 수를 먼저 뽑기 위해서는 안에 있는 요소를 다 빼면서 가장 큰 값이 무엇인지 확인해야한다. 이 과정에서 O(N), 가장 큰 값을 찾고 나머지를 다시 넣는 과정에서 O(N)의 시간 복잡도가 또 필요하다.

하지만, 우선순위 큐로 구현한 Stack의 경우에는 Push 과정에서 우선순위 큐에 Enqueue되기 때문에 이 큐는 Heap에 의해 구성되어있으므로 O(log\_2(N))의 시간복잡도가 필요하다. Pop 할 때도 마찬가지로 우선순위 큐에서 가장 큰 수가 먼저 나오기 때문에 O(log\_2(N))의 시간복잡도가 필요하다.