

제5회 연습문제

1. 지난 시간의 연습문제에 quicksort 알고리즘을 추가하라.
2. quicksort 알고리즘의 partition 함수에서 첫번째 값, 중간 위치의 값, 그리고 마지막 값 중에서 중간값을 pivot으로 선택하도록 알고리즘을 수정하여 1번 문제에 추가하라. 이미 정렬된 데이터를 입력으로 주었을 때 정렬에 걸리는 시간을 기본 quicksort 알고리즘과 비교하라.
3. Selection 문제는 n 개의 정수들 중에서 k 번째로 작은 정수를 찾는 문제이다. k 는 입력으로 주어진다. 먼저 정렬을 한 후 k 번째 정수를 찾는다면 시간복잡도는 $O(n\log n)$ 이 된다. quickselection이라고 부르는 다른 한 가지 방법은 quicksort의 partition 함수를 이용하는 것이다. 즉 먼저 주어진 정수들을 quicksort에서 처럼 하나의 값을 pivot으로 선택하여 pivot보다 작은 값들과 큰 값들로 분할 한다. pivot보다 작은 값들의 개수가 p 개라고 가정하자. 만약 $p \geq k$ 라면 그 들 중에서 k 번째로 작은 수를 recursion으로 찾는다. 만약 $p = k - 1$ 라면 pivot이 바로 k 번째로 작은 수이다. $p < k - 1$ 라면 이제 pivot보다 큰 수들 중에서 $k - p - 1$ 번째로 작은 수를 리커전으로 찾는다. 이 방법의 최악의 경우 시간복잡도는 quicksort의 경우와 마찬가지로 $O(n^2)$ 이다. 하지만 평균시간복잡도는 $O(n)$ 이다.¹ quickselection 알고리즘을 구현한 후 정렬 알고리즘과 마찬가지로 100000개의 데이터에 대해서 정렬을 한 후 k 번째 값을 찾는 방법과 비교하라. (입력에 동일한 정수들이 있는 경우: k 번째로 작은 정수는 정렬을 했을 때 k 번째에 위치하는 값을 의미함. 가령 정수들이 1, 2, 2, 2, 3, 4이고 $k=3$ 이면 2를 출력하면 됨).

¹ Selection 문제에 대해서는 최악의 경우 시간복잡도가 $O(n)$ 인 알고리즘이 존재한다.