출석수업 과제물(평가결과물) 표지(온라인제출용)

**교과목명 : 알고리즘**

**학 번 : 202334-243503**

**성 명 : 정종길**

**강 의 실 : ZOOM 8**

**연 락 처 : 010-3002-1272**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[문제 1]

* 1. O(1) < O(logn) < O(n) < O(nlogn) < O(n2) < O(n3) < O(n4)
  2. O(1) = 1  
     O(logn) = 2.1072  
     O(n) = 128  
     O(nlogn) = 269.7228  
     O(n2) = 16384  
     O(n3) = 2097152  
     O(n4) = 268435456

[문제 2]

(1) 무게 1당 이익을 구한다.

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

(2) 이 중 가장 이익이 큰 3번을 먼저 채운다.

남은 가방 공간 : 21 - 4 = 16, 이익 : 28

(3) 두번째로 이익이 큰 4번을 채운다.

남은 가방공간 : 16 - 6 = 10, 이익 : 28 + 36 = 64

(4) 세번째로 이익이 큰 5번을 채운다.

남은 가방공간 : 10 - 3 = 7, 이익 : 64 + 15 = 79

(5) 네번째로 이익이 큰 2번을 채운다.

남은 가방공간 : 7 - 4 = 3, 이익 : 79 + 16 = 95

(6) 다섯번째로 이익이 큰 1번을 채우는데, 남은 공간만큼만 채운다.

남은 가방공간 : 3 - 3 = 0, 이익 : 95 + (27/3) = 104

최대 이익 : 104

[문제 3]

(1) 먼저 이진탐색을 사용하기 위해 배열을 정렬한다. 정렬은 퀵정렬을 이용한다.

배열 크기를 반으로 나눈 인덱스 값인 35를 피벗으로 잡고 left 70과 right 20의 자리를 변경한다.

결과 : {20, 25, 35, 15, 40, 45, 70}

(2) 다음 left가 40 right가 15로 left > right이기 때문에 피벗위치를 중앙으로 이동시킨다. 배열 크기는 7로 홀수이기 때문에 중앙의 값은 7 / 2 + 7 % 2로 계산한다.

결과 : {20, 25, 15, 35, 40, 45, 70}

(3) 피벗인 35를 기준으로 좌측 우측 배열을 정렬한다. 좌측의 20을 피벗으로 잡고 left 25, right 15를 구해 위치를 바꾸어준다.

결과 : {20, 15, 25, 35, 40, 45, 70}

(4) 좌측의 배열의 중앙으로 피벗 20을 이동시켜준다.

결과 : {15, 20, 25, 35, 40, 45, 70}

(5) 우측 배열의 70을 피벗으로 잡는다. 70이 가장 큰 숫자이기때문에 그대로 두고 45를 피벗으로 잡는다. 또다시 45가 40보다 큼으로 정렬을 종료한다.

결과 : {15, 20, 25, 35, 40, 45, 70}

(6) 정렬된 배열 크기를 반으로 나눈 값을 구한다. 소수점은 버린다.

7 / 2 = 3

(7) 3번의 값인 25와 탐색기 20을 비교한다.

(8) 3번의 값이 더 크기 때문에 좌측 2개의 배열의 크기를 반으로 나눈 값을 구한다.

2 / 2 = 1

(9) 1번 값인 20과 탐색키 20을 비교하여 동일한 값을 찾아낸다.

[문제 4]

(1) 먼저 가장 왼쪽 30을 피벗으로 잡고, 피벗 + 1 위치를 left, n(배열의 수) - 1 위치를 right로 잡는다.

(2) left는 오른쪽으로 이동하며 피벗보다 큰 수를 찾고, right 왼쪽으로 이동하며 피벗보다 작은 수를 찾는다.

left는 세번째 50, right는 여섯 번째 25로 이 둘의 자리를 바꾼다.

결과 : A[] = {30, 10, 25, 45, 15, 50, 40}

(3) 다시 진행이 멈춘 지점으로부터 찾기를 진행한다.

left는 네번째 45, right는 다섯 번째 15이다. 이 둘의 자리를 바꾼다.

결과 : A[] = {30, 10, 25, 15, 45, 50, 40}

(4) 다시 진행이 멈춘 지점으로부터 찾기를 진행한다.

left는 다섯 번째 45, right는 네번째 15이다. 이 둘의 크기를 비교하여 left > right일 경우 찾기를 멈추고 피벗을 배열의 중앙으로 이동시킨다. 중앙의 위치는 7(배열 크기) / 2 + 1 인 4의 위치로 이동시킨다.

결과 : A[] = {10, 25, 15, 30, 45, 50, 40}

(5) 피벗(30)을 중심으로 좌측과 우측을 다시 정렬한다.

좌측 10을 피벗으로 삼았을 때 나머지 수가 모두 피벗보다 크기 때문에 자리에 변동을 하지 않는다.

다음 숫자인 25가 그 다음 숫자인 15보다 크기 때문에 둘의 자리를 교환한다.

결과 : A[] = {10, 15, 25, 30, 45, 50, 40}

(6) 우측 50을 피벗으로 삼았을 때 나머지 수가 모두 피벗보다 작기 때문에 자리에 변동을 하지 않는다.

다음 숫자인 40이 그 다음 숫자인 45보다 작기 때문에 둘의 자리를 바꾸어 준다.

결과 : A[] = {10, 15, 25, 30, 40, 45, 50}

(8) 더이상 정렬할 것이 없기 때문에 퀵 정렬을 종료한다.

퀵 정렬은 불안정적인 정렬이다. 왜냐하면 동일한 값이 있는 경우 처음의 순서를 보장하지 않기 때문이다. 예시로 주어진 배열이 {30(A), 30(B), 25, 45, 15, 50, 40}이라 할 때, {15, 25, 30(B),30(A),40,45,50}으로 정렬되는 것으로 확인이 가능하다.

또, 퀵 정렬을 자리를 변경하는 것뿐이기 때문에 추가로 메모리 공간이 필요하지 않기 때문에 제자리 정렬이다.

[문제 5]

퀵 정렬은 피벗을 기준으로 배열을 2개의 부분 배열로 분할하고 각 부분 배열에 대해서 퀵 정렬을 순환적으로 적용하는 것이고, 합병 정렬은 주어진 배열을 동일한 크기의 2개의 부분배열로 분할하여 각 부분배열을 순환적으로 합병 정렬을 적용하여 정렬시킨 후 정렬된 두 부분 배열을 합병하여 하나의 정렬된 배열로 만드는 방식이다.

퀵 정렬과 합병 정렬 둘 모두 분할 정렬 방법이 적용된 것이며, 차이점으로는 세 가지가 있다.

2. 퀵 정렬은 제자리 정렬이지만 합병 정렬은 전체적으로 입력크기 n만큼의 추가적인 저장공간이 필요한 알고리즘

3. 퀵 정렬은 동일한 값의 처음 순서를 보장하지 못하는 불안정적 정렬이지만 합병 정렬은 안정적 정렬

4. 합병 정렬과 다르게 퀵 정렬은 배열의 순서에 따라 성능이 차이가 난다.

이 중 3번에 기술한 성능에 대하여 설명하면 합병 정렬은 항상 일정하게 나누어 정렬 후 merge 하기때문에, 각각 데이터 정렬 시간 + merge의 식으로 나타낼 수 있다.

n개의 데이터를 각각 나누어 정렬하는 시간을 더하면 T(n/2) + T(n/2) 이 된다.

merge하는 시간은 두 부분 배열의 비교 횟수는 최소 n/2, 최대 n-1이므로 데이터 개수 n에 비례하는 형태가 되어 Θ(n)이 되므로, 적용하면 2T(n/2)+Θ(n)이된다.

이를 점화식으로 표현하면

T(n)=O(nlogn)이 된다.

퀵 정렬은 먼저 분할 함수의 성능을 구한다. 각 데이터와 피벗과의 비교 횟수를 도출하면 1회 많으면 2회 비교가 된다. 즉, 데이터에 비례하는 횟수만큼 이기때문에 Θ(n)이다.

피벗에 의해 나누어진 좌측과 우측 정렬 시간을 구하여 분할 함수 성능을 더해주면 되는데, 불균형 하게 나누어 지므로,

T(n) = T(nL) + T(nR) + Θ(n)로 나타낼 수 있다.

분할된 두 부분 배열의 크기의 경우의 수를 나타내면 n가지가 만들어지는데, 그 중 피벗만 제자리를 잡고 나머지 원소가 하나의 부분 배열이 되는 0:n-1, n-1:0이 퀵 정렬의 최악의 성능으로, 시간 복잡도는

T(n) = T(n-1) + T(0) + Θ(n)

T(n) = T(n-1) + Θ(n)

T(n) = O(n2)이 된다.

다음 최선의 성능은 항상 n/2 : n/2로 나누어지는 경우로, 시간 복잡도는

T(n) = T(n/2) + T(n/2) +Θ(n)

T(n) = 2T(n/2)+Θ(n)

T(n)=O(nlogn)이 된다.

최선과 최악이 있는 퀵 정렬의 평균 수행 시간은 모든 분할 비율(경우)에 따라 수행 시간의 평균을 낸 값으로 T(n)=O(nlogn)이 된다.

※ 표지는 A4용지 사용