

1. 소스코드 및 실행결과

- Exchange Sort

```
# 교환정렬(exchange sort)
def exchangeSort(data):
    for i in range(0, len(data) - 1):
        for j in range(i + 1, len(data)):
            if data[i] > data[j]: # 인덱스 위치를 바꿔준다.
                data[i], data[j] = data[j], data[i]
    return data

for i in [2000, 4000, 12000]:
    start = time.time()
    n = i # 배열의 수
    data = [] # 무작위 배열
    for _ in range(n):
        data.append(random.randint(1, 100))

    exchangeSort(data)

    print("n = " + str(i) + " 일 때 걸린 시간 : ", time.time() - start)
```

executed in 5.96s, finished 18:02:00 2021-03-11

n = 2000 일 때 걸린 시간 : 0.16051769256591797
 n = 4000 일 때 걸린 시간 : 0.6053805351257324
 n = 12000 일 때 걸린 시간 : 5.179145812988281

➔ 교환정렬(exchange sort)의 시간복잡도는 $O(n^2)$ 에 비례하며 $n = 2000$ 일 때 약 4000000번의 연산을 수행하며 대략 0.1605초가 걸린다. 그러므로 $\{x : 60 = 4,000,000 : 0.1605\}$ 의 비례식을 풀어 x (1분 간 연산의 횟수)를 구하면 대략 1,495,327,102.8번의 연산 횟수가 도출된다. 이 연산 횟수에 루트를 취하면 약 38,669이 도출된다. 그러므로 해결할 수 있는 문제의 크기 n 은 약 38669이다.

- Merge Sort

```

#합병정렬(merge sort)

def mergeSort(data):
    if len(data) <= 1:
        return data

    mid = len(data) // 2

    left = data[:mid]
    right = data[mid:]

    left = mergeSort(left) # 앞쪽
    right = mergeSort(right) # 뒤쪽

    return merge(left, right)

def merge(left, right):
    i = j = 0
    sub_data = []
    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            sub_data.append(left[i])
            i += 1
        else:
            sub_data.append(right[j])
            j += 1

    while i < len(left):
        sub_data.append(left[i])
        i += 1
    while j < len(right):
        sub_data.append(right[j])
        j += 1
    return sub_data

for i in [2000, 4000, 12000]:
    start = time.time()
    n = i # 배열의 수
    data = [] # 무작위 배열
    for _ in range(n):
        data.append(random.randint(1, 100))

    mergeSort(data)

    print("n = " + str(i) + " 일 때 걸린 시간 : ", time.time() - start)

```

executed in 127ms, finished 18:03:49 2021-03-11

n = 2000 일 때 걸린 시간 : 0.011969327926635742
n = 4000 일 때 걸린 시간 : 0.02396106719970703
n = 12000 일 때 걸린 시간 : 0.07280969619750977

➔ 합병정렬(merge sort)의 시간복잡도는 $O(n\log n)$ 에 비례한다. $n = 2000$ 일 때 약 6,602번의 연산을 수행하며 약 0.0119초가 걸린다. 그러므로 $\{x : 60 = 6,602 : 0.0119\}$ 의 비례식을 풀어 x 를 구하면 대략 33,287,395번의 연산 횟수가 도출되며 이 값을 " $n\log n = 33,287,395$ " 식으로 n 을 구하면 됩니다. n 의 값이 너무 커서 계산기로도 값이 구해지지 않습니다..

건의사항 : 가능하다면 과제 데드라인 이후 풀이 및 정답을 공유해주셨으면 좋겠습니다. 감사합니다.