

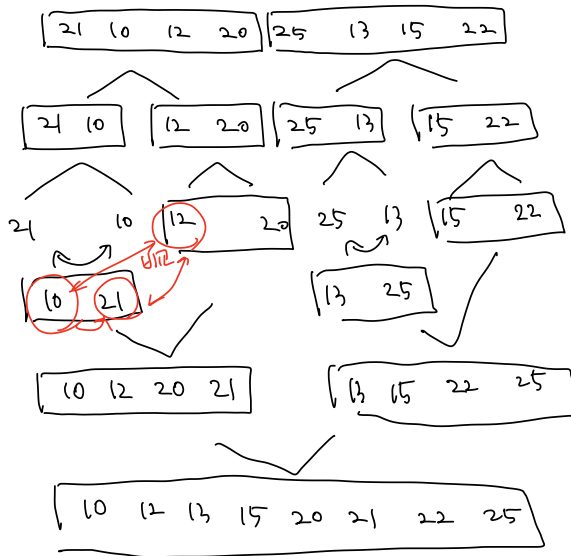
- 문제 2: Merge Sort, 크기 n 인 배열을 입력으로 받아,

배열을 절반으로 두개로 나눈 후,

각 작은 배열을 재귀적으로 정렬하고,

그 결과를 Merge한다. Mergesort의 수도 코드를 간략하게 작성해보고 시

간 복잡도를 증명하시오.



Merge Sort 부분: 비교연산, 이동연산 X

Merge 부분:

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$T(1) = 1$$

$$\frac{T(2)}{2} = \frac{T(1)}{2} + 1$$

$$\frac{T(4)}{4} = \frac{1}{2} \frac{T(2)}{2} + 1$$

$$\frac{T(n)}{n} = \frac{2}{n} \cdot \frac{T\left(\frac{n}{2}\right)}{2} + 1$$

$$\frac{T(n)}{n} = \log_2 n + 1$$

$$T(n) = n \log n + n$$

$$O(n \log n)$$

Merge Sort (array, start, end):

if start < end:

$$mid = (start + end) / 2$$

Merge Sort (array, start, mid)

Merge Sort (array, mid + 1, end)

Merge (array, start, mid, end)

↳ Merge (array, start, mid, end):

$$n_1 = mid - start + 1$$

$$n_2 = end - mid$$

Let $L[1:n_1+1]$ and $R[1:n_2+1]$

for i in range $(1, n_1+1)$

$$L[i] = array[start + i - 1]$$

for j in range $(1, n_2+1)$

$$R[j] = array[mid + j]$$

$$L[n_1+1] = \infty, R[n_2+1] = \infty$$

$$i = 1, j = 1$$

for k in range $(start, end+1)$

if $L[i] \leq R[j]$

$$A[k] = L[i]$$

$$i = i + 1$$

else $A[k] = R[j]$

$$j = j + 1$$

- 문제 4: 위의 소팅 알고리즘에서 수행하는 Swap의 횟수는 최대 몇 번인가?

$$n \log_2 n$$

- 문제 6: 루트 있는 트리를 입력으로 받아 아래와 같이 출력하는 알고리즘을 작성하라.

트리의 각 노드에는 1,000 미만의 자연수가 저장되어 있다. 트리의 노드 연결 관계는 다음과 같이 표현해야 한다. 아래 출력에서 루트에는 자식이 3개 있고 그 자식들 중 하나는 더 이상 자식이 없는 것임을 알 수 있을 것이다.

```
[030]---[054]-----[001]
      +---[002]
      L--[045]-----[123]
```

```
Tree = []
```

```
Tree.append([node, parent, *child])
```

```
for i in range (len(tree)):
```

```
    if tree[i][1] == 0;
```

```
    pre_order[i]
```

def
→

```
        print(node)
```

```
        print('--')
```

```
        if child:
```

```
            for j in len(child)
```

```
                if j != len(child)
```

```
                    print(++)
```

```
                    pre_order[tree[i][2+j]]
```

```
        else:
```

```
            print(L--)
```

```
            pre_order[tree[i][2+j]]
```