

Data Structure

Week 15
KyuDong SIM

1. 이번 주 실습 내용

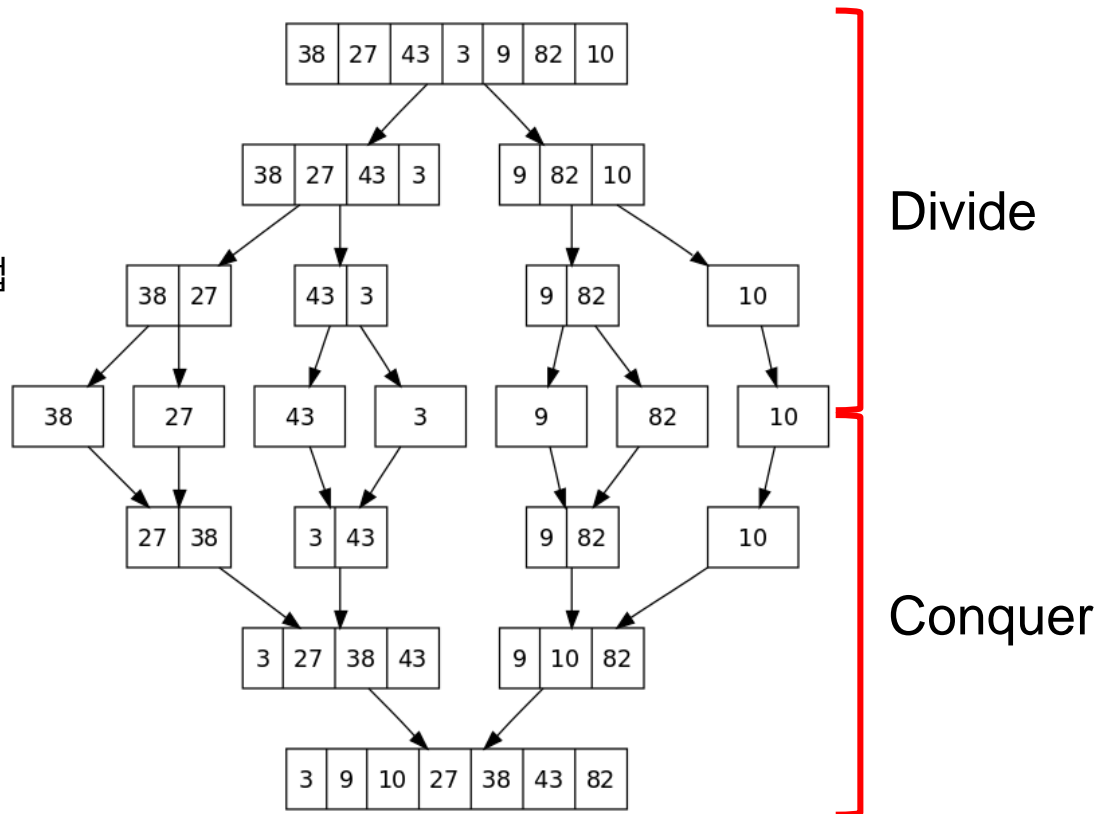
- Merge Sort 구현

Merge Sort

- Divide and Conquer

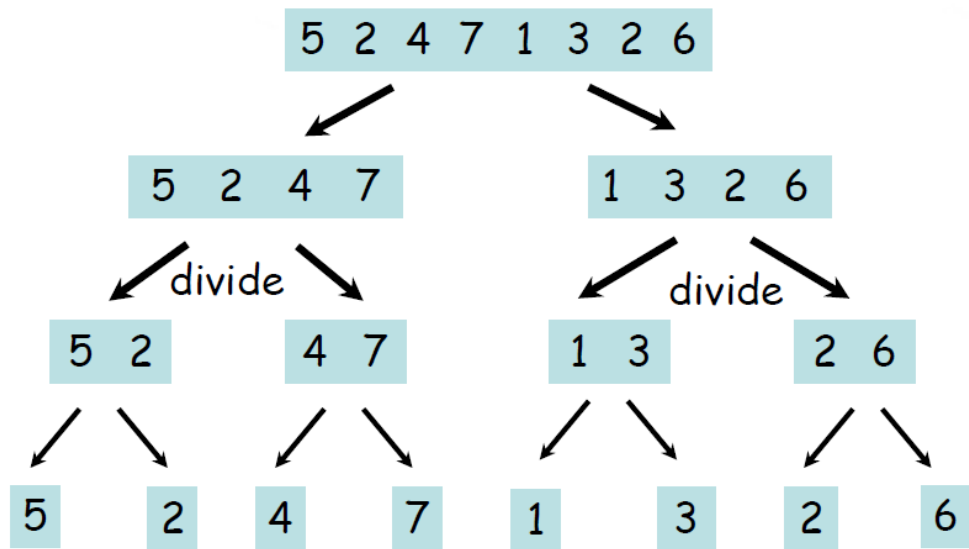
작게 분할하여 문제를 해결하는 방법

- Recursive



Merge Sort

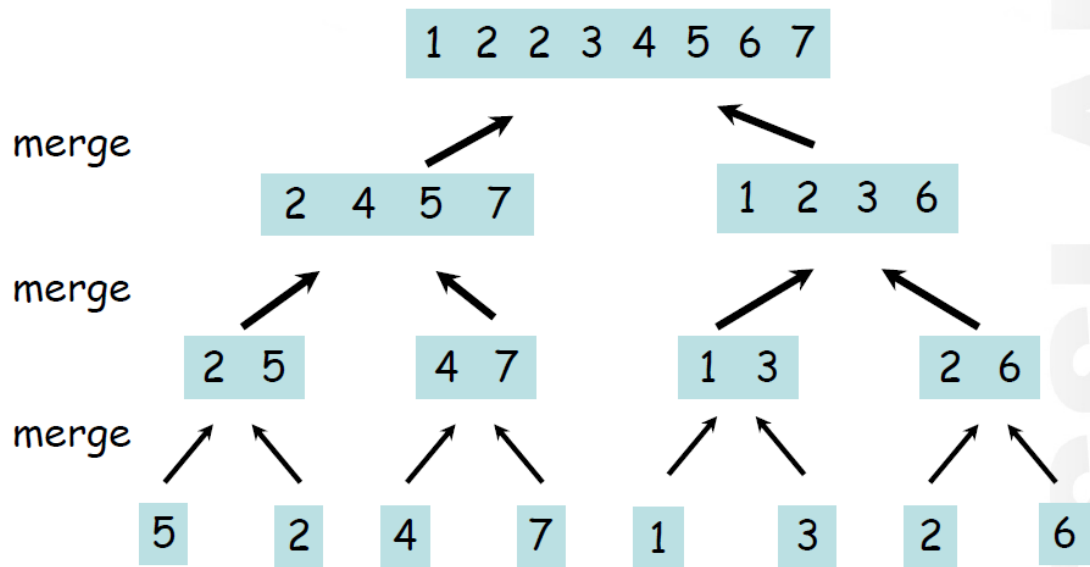
- Divide



Dong Kyue Kim: Algorithm

Merge Sort

- Conquer



Dong Kyue Kim: Algorithm

Merge Sort

- Merge Algorithm

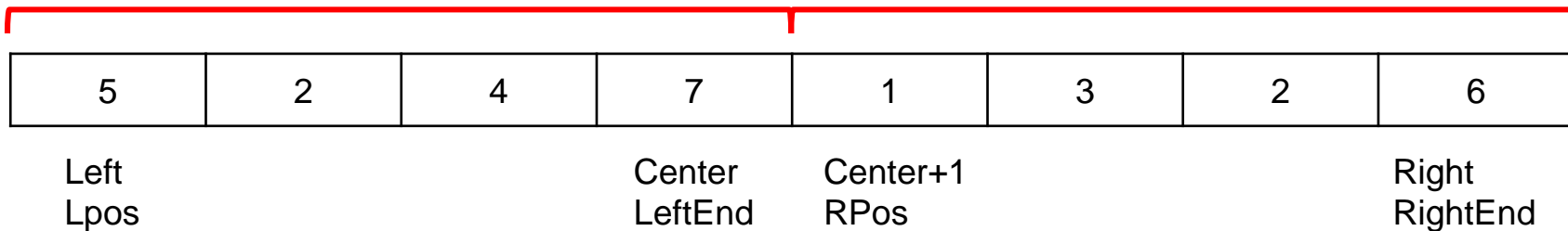
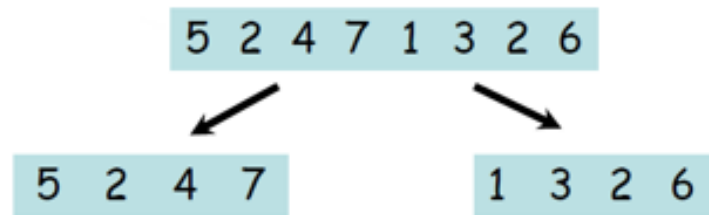
- $\langle 1, 5, 6, 8 \rangle \langle 2, 4, 7, 9 \rangle \rightarrow \langle 1 \rangle$
- $\langle 5, 6, 8 \rangle \langle 2, 4, 7, 9 \rangle \rightarrow \langle 1, 2 \rangle$
- $\langle 5, 6, 8 \rangle \langle 4, 7, 9 \rangle \rightarrow \langle 1, 2, 4 \rangle$
- $\langle 5, 6, 8 \rangle \langle 7, 9 \rangle \rightarrow \langle 1, 2, 4, 5 \rangle$
- $\langle 6, 8 \rangle \langle 7, 9 \rangle \rightarrow \langle 1, 2, 4, 5, 6 \rangle$
- $\langle 8 \rangle \langle 7, 9 \rangle \rightarrow \langle 1, 2, 4, 5, 6, 7 \rangle$
- $\langle 8 \rangle \langle 9 \rangle \rightarrow \langle 1, 2, 4, 5, 6, 7, 8 \rangle$
- $\langle \rangle \langle 9 \rangle \rightarrow \langle 1, 2, 4, 5, 6, 7, 8, 9 \rangle$

두 배열 중 작은 값을 새 배열에 넣는다.

Dong Kyue Kim: Algorithm

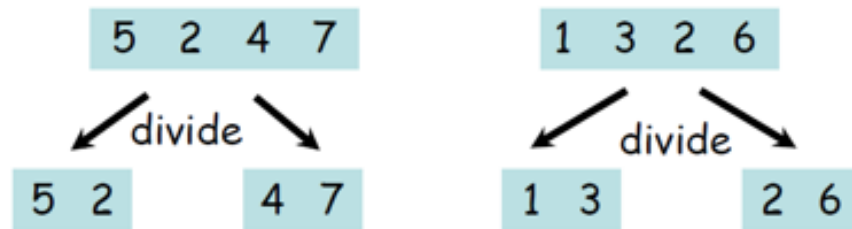
배열로 표현한 Merge Sort

- Divide를 배열과 index를 이용해 표현한다.



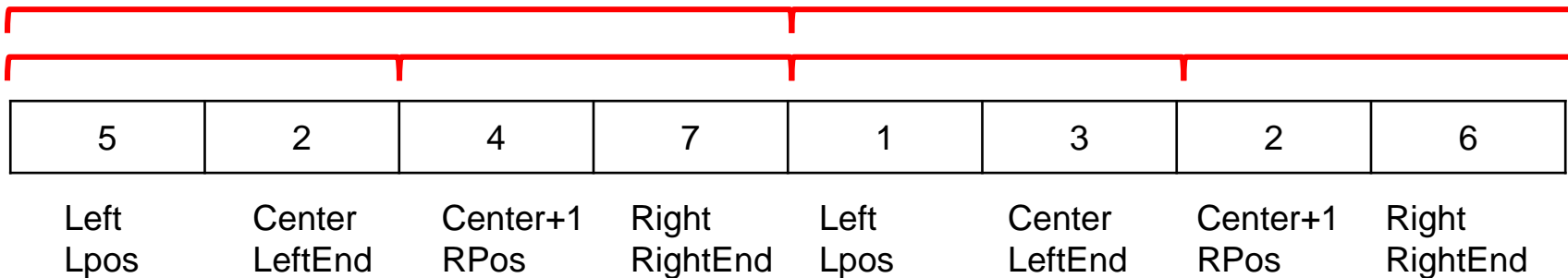
배열로 표현한 Merge Sort

- Recursive 함수에 의해 나뉜다.



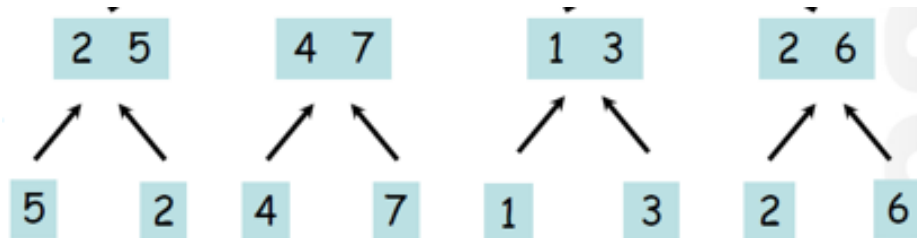
MSort1

MSort2



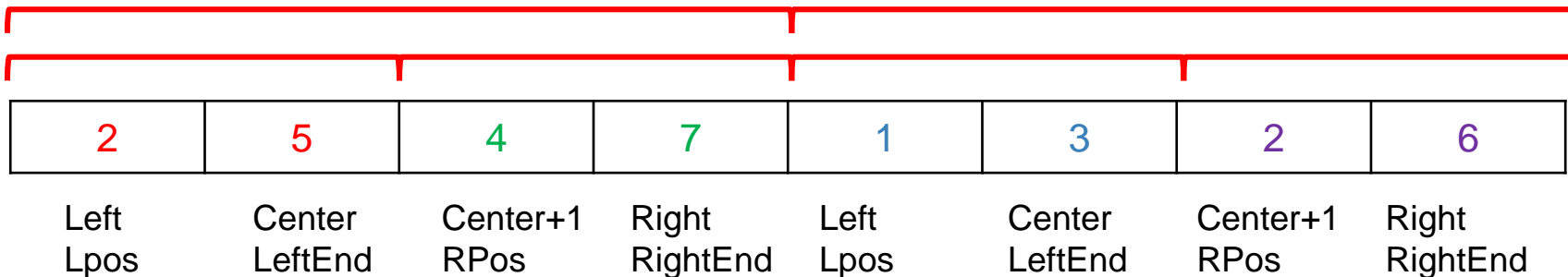
배열로 표현한 Merge Sort

- Merge에 의해 정렬되면서 합쳐짐



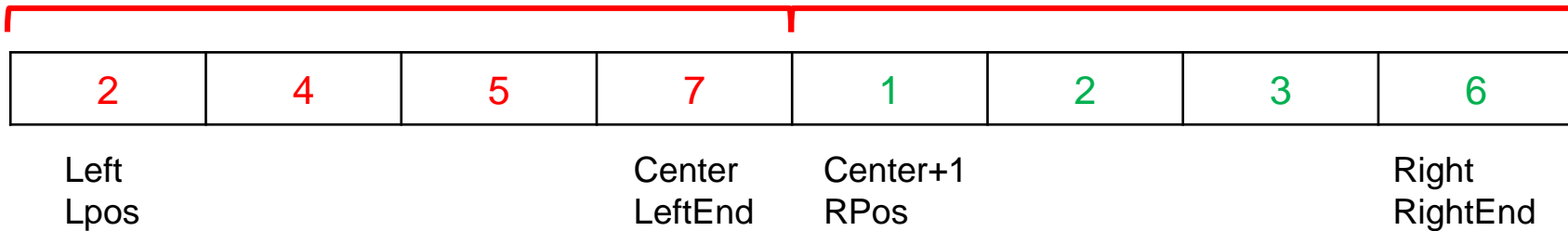
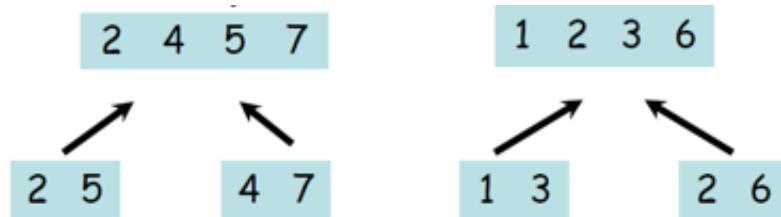
MSort1

MSort2



배열로 표현한 Merge Sort

- Merge에 의해 정렬되면서 합쳐짐



배열로 표현한 Merge Sort

- Merge에 의해 정렬되면서 합쳐짐



1	2	2	3	4	5	6	7
Left Lpos			Center LeftEnd	Center+1 RPos			Right RightEnd

MSort

```
void MSort (ElementType A[], ElementType TmpArray[ ], int Left, int Right)
{
    int Center;
    if (Left < Right){
        Center = (Left + Right) / 2;
        MSort (A, TmpArray, Left, Center);
        MSort (A, TmpArray, Center+1, Right);
        Merge (A, TmpArray, Left, Center+1, Right);
    }
}
```

배열 크기가 1이 될
때까지 Recursive

반으로 나눠 Msort를
recursive 함수로
불러옴

Merge 실행

Merge

```
void Merge (ElementType A[], ElementType TmpArray[ ], int Lpos, int Rpos, int RightEnd)
{
    int i, LeftEnd, NumElements, TmpPos;
    LeftEnd = Rpos - 1;
    TmpPos = Lpos;
    NumElements = RightEnd - Lpos + 1;

    while (Lpos <= LeftEnd && Rpos <= RightEnd)
        if (A[Lpos] <= A[Rpos])
            TmpArray[TmpPos++] = A[Lpos++];
        else
            TmpArray[TmpPos++] = A[Rpos++];

    while (Lpos <= LeftEnd)
        TmpArray[TmpPos++] = A[Lpos++];
    while (Rpos <= RightEnd)
        TmpArray[TmpPos++] = A[Rpos++];

    for(i=0; i<NumElements; i++, RightEnd--)
        A[RightEnd] = TmpArray[RightEnd];
}
```

입력 배열 A

비어있는 배열 TmpArray

Left 와 Right 둘 중 하나가 끝날
때까지 작은 값을 TmpArray에
대입

남은 Left 또는 Right를
TmpArray에 마저 채움

정렬된 TmpArray 값을
A행렬에 대입

결과 예

```

C:\Windows\system32\cmd.exe
Before Sorting :    5   12    1    3    2   11    8    9    7    4    6   10
Safter Sorting :    1    2    3    4    5    6    7    8    9   10   11   12
계속하려면 아무 키나 누르십시오 . . .
  
```

제출 및 알림

수업 중 확인 or 메일제출 (학번 써주세요)

메일 제출 :

주소 : (89kdsim@naver.com)

기한 : ~2016-06-15