

# 분할 정복 2

이충기

명지대학교  
컴퓨터공학과

# 문제 1: 재배치

(50점)  $n (\geq 1)$  개의 정수들의 배열이 주어  
져 있다. 이 배열의 요소들을 모든 음수(0보  
다 작은 정수) 요소들이 모든 양수(0보다 크  
거나 같은 정수) 요소들보다 앞에 나오도록  
재배치하는  **$O(n)$**  알고리즘을 작성하라. (힌  
트: 빠른 정렬의 분할 알고리즘을 참고하라.)

# ① 문제1

```
public static void sort(int A[], int len){ // 정렬할 배열과 배열의 크기를 입력받는다.
    int i = 0; // 1. 처음 시작 부분을 저장하는 정수 변수 i를 선언한다.
    int j = len - 1; // 2. 배열의 끝 부분 인덱스를 저장하는 정수 변수 j를 선언한다.
    while(i <= j){ // 3. j가 i이상일 경우 :
        if(A[i] <= 0){ // A[i]가 0 이하면 i를 1 증가시킨다.
            i = i + 1;
        }
        else if(A[j] > 0){ // A[j]가 0 이하면 j를 1 감소시킨다.
            j = j - 1;
        }
        else{ // 그 외에는 A[i]와 A[j]의 값을 바꾸고(지역 변수 temp를 선언하고 A[i]를 저장하고, A[i]에 A[j] 값을 넣은 뒤 A[j]에 temp값을 넣는다.)
            // i를 1 증가, j를 1 감소시킨다.
            int temp = A[i];
            A[i] = A[j];
            A[j] = temp;
            i = i + 1;
            j = j - 1;
        }
    }
    // 4. j가 i보다 작을 경우는 반복문 종료
}
```

## 문제 2: 토끼 수 구하기

(암수) 한 쌍의 새로 태어난 토끼들은 한 달이 될 때에 두 (암수) 쌍의 토끼를 낳고 두 달이 될 때에 또 다른 세 (암수) 쌍의 토끼를 낳는다고 가정한다. 내가 방금 한 쌍의 새로 태어난 토끼들을 샀다면 지금부터  $n$  번째 달에 몇 쌍의 토끼들이 태어나는 지를 계산해야 한다.

1. (6점) 지금부터  $i$  번째 달에 새로 태어난 토끼 쌍들의 수들로 이루어진 수열의 일부 항을 구하라.  $a_i$ 를  $i$  번째 달에 새로 태어난 토끼 쌍들의 수라고 하면  $a_0 = 1, a_1 = 2, a_2 = 7$  이다.  $a_3, a_4$ 와  $a_5$ 를 구하라.
2. (14점) 1에서 구한 수열을 나타내는 점화식을 작성하라.
3. (15점) 2에서 구한 점화식을 이용하여  $n$  이 주어지면  $a_n$ (새로 태어난 토끼 쌍들의 수)를 구하는 비재귀 알고리즘(반복문 사용)을 작성하라. 작성한 알고리즘의 시간복잡도를 O-표기로 나타내라.
4. (15점) 2에서 구한 점화식을 이용하여  $n$  이 주어지면  $a_n$ (새로 태어난 토끼 쌍들의 수)를 구하는 재귀 알고리즘을 작성하라.

다크 야인  $a_0 = 1, a_1 = 2, a_2 =$

...

## ②문제 2

$$2-1. \quad a_3 = 2 \times 3 + 7 \times 2 = 6 + 14 = 20$$

$$a_4 = 7 \times 3 + 20 \times 2 = 21 + 40 = 61$$

$$a_5 = 20 \times 3 + 61 \times 2 = 60 + 122 = 182$$

2-2. 점화식은

$$a_i \begin{cases} i=0 & \rightarrow 1 \\ i=1 & \rightarrow 2 \\ i>1 & \rightarrow a_i = a_{i-1} \times 2 + a_{i-2} \times 3 \end{cases}$$

2-3. 시간 복잡도는  $O(n)$

```
public static int rabbit1(int n){ // n을 입력받는다.
    int A[] = new int[n + 1]; // 1. n + 1 크기의 배열을 선언한다.
    if (0 <= n) { // 2. 만약 n이 0 이상이라면 A[0]에 1을 넣는다.
        A[0] = 1;
    }
    if (1 <= n) { // 3. 만약 n이 1 이상이라면 A[1]에 2를 넣는다.
        A[1] = 2;
    }
    for(int i = 2; i <= n; i++){ // 3. 지역변수 i를 선언하고 i가 2에서 n까지 증가할 때 A[i] = A[i - 1] * 2 + A[i - 2] * 3
        A[i] = A[i - 1] * 2 + A[i - 2] * 3;
    }
    return A[n]; // 4. 반복문이 모두 끝나면 A[n]을 반환한다.
}
```

2-4.

```
public static int rabbit2(int n){ // n을 입력받는다.
    if(n == 0 || n == 1){ // 1. n이 0일 때는 1을, n이 1일 때는 2를 반환한다.
        return n + 1;
    }
    else { // 2. 그 외에는 rabbit2(n - 1) * 2 + rabbit2(n - 2) * 3를 반환한다.
        return rabbit2(n - 1) * 2 + rabbit2(n - 2) * 3;
    }
}
```