

2024 겨울학기 동국대학교SW역량강화캠프

12일차. 정렬과 분할정복



● 정렬

- ▶ 일정 기준에 따라서 원소들을 순서대로 나열하는 과정
- ▶ 버블 정렬, 선택 정렬, 삽입 정렬, 힙 정렬, 병합 정렬, 퀵 정렬, 계수 정렬, 기수 정렬
- ▶ Java에서는 `Array.sort()` 나 `Collection.sort()` 와 같은 정렬 알고리즘이 내장

- ▶ 버블 정렬, 삽입 정렬, 병합 정렬

● 버블 정렬

- ▶ 인접한 두 원소끼리 비교하여 정렬하는 알고리즘
- ▶ 한 번 선형 탐색 할 때마다, 마지막 원소를 고정시킬 수 있다.
- ▶ (N-1)번 선형 탐색하여 정렬
- ▶ 시간복잡도 $O(N^2)$

● 버블 정렬

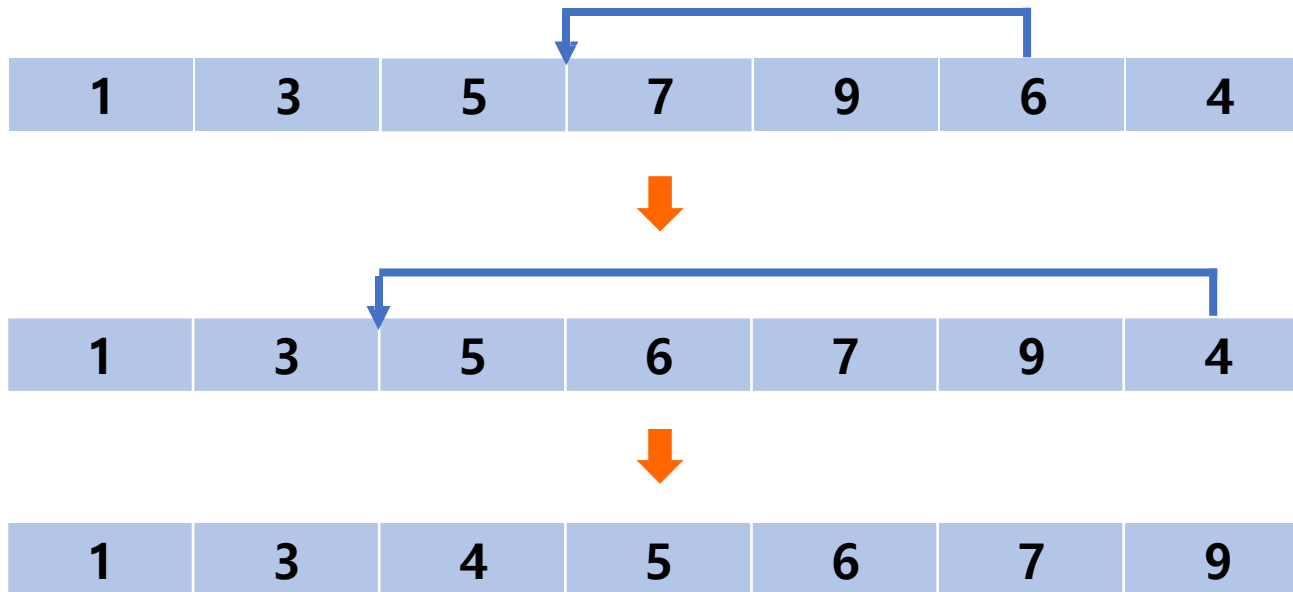
6 5 3 1 8 7 2 4

버블 정렬 코드 $O(N^2)$

```
for(int i=0; i<n; i++){  
    for(int j=0; j<n-i-1; j++){  
        if(arr[j] > arr[j+1]){  
            int tmp = arr[j];  
            arr[j] = arr[j+1];  
            arr[j+1] = tmp;  
        }  
    }  
}
```

● 삽입 정렬

▶ 원소를 정렬된 배열에서 위치를 찾아 삽입하여 정렬하는 알고리즘



▶ 시간복잡도 $O(N^2)$

● 삽입 정렬

6 5 3 1 8 7 2 4

삽입 정렬 코드 $O(N^2)$

```
for(int i=0; i<n; i++){  
    for(int j=i; j>0; j--){  
        if(arr[j] < arr[j-1]){  
            int tmp = arr[j];  
            arr[j] = arr[j-1];  
            arr[j-1] = tmp;  
        }  
    }  
}
```


● 병합 정렬

▶ 배열을 2등분하여 각각 정렬하고, 정렬된 두 배열을 병합하는 것을 재귀적으로 진행하여 배열을 정렬하는 알고리즘

1	4	6	7	2	3	5	8
---	---	---	---	---	---	---	---



1	2	3	4	5	6	7	9
---	---	---	---	---	---	---	---

● 병합 정렬

6 5 3 1 8 7 2 4

● 병합 정렬

- ▶ 구간의 길이가 8이라면 병합 Layer = 3 (1 → 2 → 4 → 8)
- ▶ 구간의 길이가 16이라면 병합 Layer = 4 (1 → 2 → 4 → 8 → 16)
- ▶ 시간복잡도 = $O(\text{Layer 수} * N) = O(N \lg N)$
- ▶ 컴퓨터 과학에서 $\lg N = \log_2 N$

병합 정렬 코드 $O(N \lg N)$

```
static void merge_sort(int[] arr, int l, int r){
    if(r-l == 1) return;

    int m = (l+r)/2;
    merge_sort(arr, l, m);
    merge_sort(arr, m, r);

    int lp=l, rp=m;
    int[] tmp = new int[r];
    for(int i=l; i<r; i++){
        if(rp == r || (lp < m && arr[lp] < arr[rp])) tmp[i] = arr[lp++];
        else tmp[i] = arr[rp++];
    }

    for(int i=l; i<r; i++) arr[i] = tmp[i];
}
```

● 하노이의 탑(5342)

세 개의 장대가 있고 첫 번째 장대에는 반경이 서로 다른 n 개의 원판이 쌓여 있다. 각 원판은 반경이 큰 순서대로 쌓여있다. 이제 수도승들이 다음 규칙에 따라 첫 번째 장대에서 세 번째 장대로 옮기려 한다.

한 번에 한 개의 원판만을 다른 탑으로 옮길 수 있다.

쌓아 놓은 원판은 항상 위의 것이 아래의 것보다 작아야 한다.

이 작업을 수행하는데 필요한 이동 순서를 출력하는 프로그램을 작성하라. 단, 이동 횟수는 최소가 되어야 한다.

- ▶ 분할 정복을 이용한 풀이
- ▶ 크기 N인 원판을 1번에서 3번으로 옮기기 위해서는, N번보다 작은 원판들은 전부 2번에 있어야만 한다.

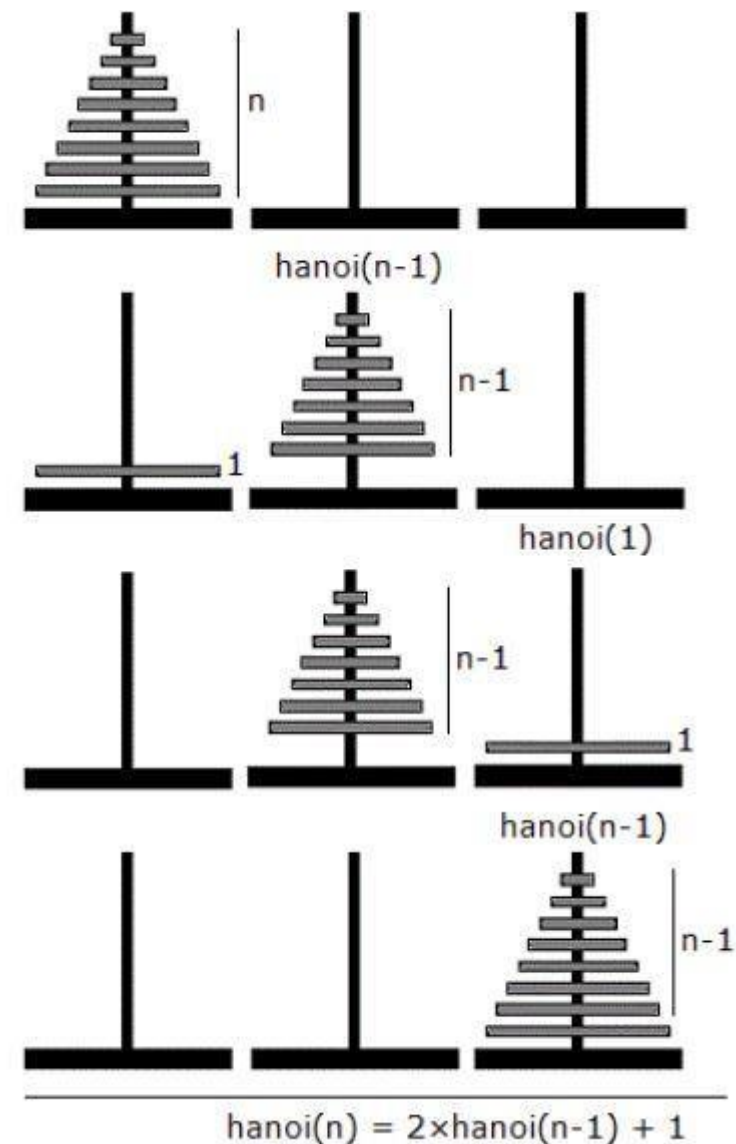
hanoi(from, to, N) 은

hanoi(from, other, N-1)

move(from, to, N)

Hanoi(other, to, N-1)

의 세 단계로 나누어 생각할 수 있다.



```
public static void main(String[] args) throws IOException{
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

    int N = Integer.parseInt(br.readLine());

    System.out.println((1<<N) - 1);
    hanoi(N, 1, 3);
}

static void hanoi(int n, int start, int end){
    if(n == 0) return;

    int mid = 6 - start - end;

    hanoi(n-1, start, mid);
    System.out.println(start + " " + end);
    hanoi(n-1, mid, end);
}
```

● 분할정복으로 별 찍기(5341)

재귀적인 패턴으로 별을 찍어 보자. N 이 3의 거듭제곱(3, 9, 27, ...)이라고 할 때, 크기 N 의 패턴은 $N \times N$ 정사각형 모양이다.

크기 3의 패턴은 가운데에 공백이 있고, 가운데를 제외한 모든 칸에 별이 하나씩 있는 패턴이다.

* *

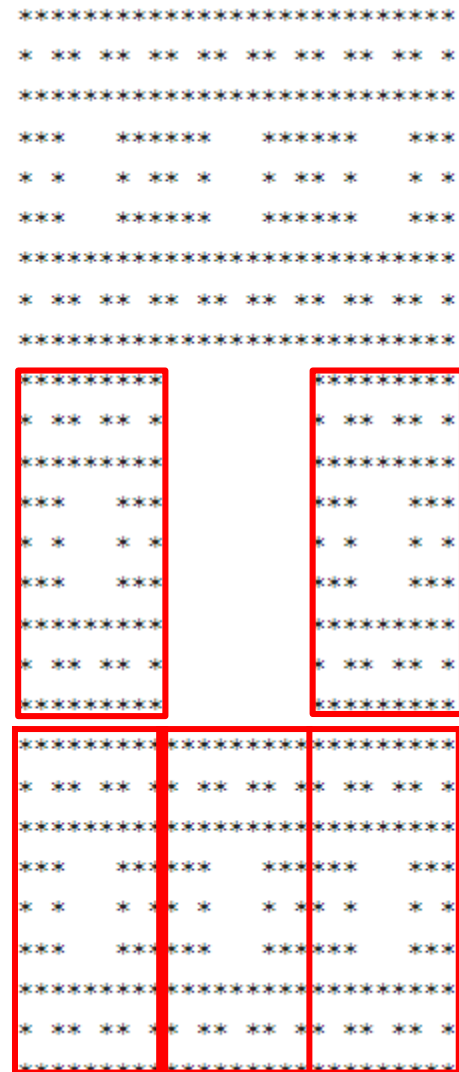
N 이 3보다 클 경우, 크기 N 의 패턴은 공백으로 채워진 가운데의 $(N/3) \times (N/3)$ 정사각형을 크기 $N/3$ 의 패턴으로 둘러싼 형태이다. 예를 들어 크기 27의 패턴은 예제 출력 1과 같다.

- ▶ 분할 정복을 이용한 풀이
- ▶ 크기 N의 패턴은 8개의 크기 (N-1)의 패턴들로 구성

$\text{star}(x, y, \text{len}) = (x, y)$ 부터 길이 len의 정사각형을 패턴으로 채우는 함수

$\text{star}(x + (\text{len}/3) * i, y + (\text{len}/3) * j, \text{len}/3) \quad (i \neq 1 \parallel j \neq 1)$

의 총 8개의 작은 부분으로 나누어 생각할 수 있다.



```
static void func(int x, int y, int l){
    if(l == 1){
        arr[x][y] = true;
        return;
    }

    for(int i=x; i<x+l; i+=l/3){
        for(int j=y; j<y+l; j+=l/3){
            if(i == x+l/3 && j == y+l/3) continue;
            func(i, j, l/3);
        }
    }
}
```

```
int N = Integer.parseInt(br.readLine());

arr = new boolean[N][N];
func(0, 0, N);

for(int i=0; i<N; i++){
    for(int j=0; j<N; j++){
        System.out.print(arr[i][j] ? '*' : ' ');
    }
    System.out.println();
}
```