

Algorithms_ch01

Sequential Search Algorithm

□ Sequential Search Algorithm

Inputs: int $n(>0)$, $S[1..n]$, x

Output: location of x in S ,
0 if not found

```
public static index SeqSearch(int n,
                              keyType[] S,
                              keyType x)
{
    index location = 1;
    while (location <= n && )
        location++;
    if (location > n)
        ;
    return location;
}
```

1. $s[\text{location}] \neq x$
2. return 0

Binary Search Algorithm

□ Binary Search Algorithm

Inputs: int $n(>0)$, $S[1..n]$, x

Output: location of x in S ,
0 if not found

```
public static index BinSearch(int n, keyType[] S,
                              keyType x)
{
    index location, low, high, mid;

    low = 1; high = n; location = 0;
    while (low <= high && location == 0) {
        mid = floor( (low+high) / 2 );
        if ( x == S[mid] )
            
        else if ( x < S[mid] )
            high = mid - 1;
        else
            
    }
    return location;
}
```

- 1.
- 2.

피보나치 수열 시간 복잡도

1. 재귀호출 $\rightarrow O(n^2)$
2. DP $\rightarrow O(n)$

Complexity - 복잡도

1. Parameters

복잡도를 결정하는 두가지 Parameters가 있다.

1. 알고리즘의 Input Size
2. 알고리즘의 Basic Operation(기본 연산)

2. Time Complexitiy

시간 복잡도는 Basic Operation이 Input size에 의한 값을 위해서 얼마나 많이 수행되는가에 따라 결정된다.

3. Intuition

결국 $N(\text{input size})$ 가 커지면 알고리즘의 복잡도는 함수에서 가장 높은 차수에 의해서 결정된다.

1. $f(n) \in \text{Big}(g(n))$

위는 $f(n) \leq c * g(n)$, $n > N$ 일 때, 부등식을 만족하게 하는 어떤 실수 c 와 N 이 있다면 성립한다. \rightarrow 위 부등식을 만족하는 어떤 값 하나만 있어도 성립한다. (모든 실수 c 에 대해서 만족하는 건 $\text{Small}(O)$ 이다.



빅오는 간단하게 차수로 따졌을 때 $g(n)$ 의 차수가 $f(n)$ 보다 높으면 성립한다.
 \rightarrow 차수가 높으면 궁극적으로 어떤 지점 N 이후로 부터 항상 높기 때문.

Ex) $f(n) = 5n$ 이고 $g(n) = n^2$ 일 때 빅오가 만족하는가?

$c = 5$ 라고 가정하면 $n \geq 1$ 일때부터 $f(n) \leq 1 * g(n)$ 이 항상 만족한다. \rightarrow 빅오는 성립한다. $\rightarrow \text{Big}(O)$ is True

2. $f(n) \in \Omega(g(n))$ - 오메가

위는 $f(n) \geq c * g(n)$, $n > N$ 일 때, 부등식을 만족하게 하는 어떤 실수 c 와 N 이 있다면 성립한다.



오메가 오는 $g(n)$ 의 차수가 $f(n)$ 보다 낮으면 성립한다.
 \rightarrow 차수가 낮으면 궁극적으로 어떤 지점 N 이후로부터 항상 작기 때문.

Ex1) $f(n) = n^2$ 이고 $g(n) = 5n$ 일 때 빅오가 만족하는가?

$c = 1/5$ 라고 가정하면 $n \geq 1$ 일 때부터 $f(n) \geq 1 * g(n)$ 이 항상 만족한다. \rightarrow 오메가 오는 성립한다. $\rightarrow \Omega(O)$ is True

Ex2) $f(n) = n^3$ 이고 $g(n) = 100n^2$ 일 때 오메가 오가 만족하는가?

c 가 높은 값 100 이라고 가정을 해도, n 이 10000이 넘어가면 차수에 의해서 언젠간 $f(n)$ 이 더 커진다. 그러므로 $f(n) \geq c * g(n)$ 은 성립한다. \rightarrow 오메가 오는 성립한다. $\rightarrow \Omega(O)$ is True

3. $f(n) \in \theta(g(n))$ - 세타

$f(n)$ 과 $g(n)$ 이 빅오와 오메가 오를 성립하면 세타는 성립한다.

For a given complexity function $f(n)$,
 $\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$

4. $f(n) \in \text{Small}(g(n))$ - 스몰 오

Big O는 어떤 하나의 실수 c 만 만족하면 된다.

Small O는 모든 실수 c 에 대해서 만족해야 한다. → 절대적인 크를 표현하기 위함.

$n \in o(n^2)$ Yes

$n \in o(5n)$? No

같은 차수면 스몰 오는 성립하지 않는다.

5. Properties of Order

1.4 Order

□ Properties of Order

1. $g(n) \in O(f(n))$ iff $f(n) \in \Omega(g(n))$
2. $g(n) \in \Theta(f(n))$ iff $f(n) \in \Theta(g(n))$
3. For $b > 1$ and $a > 1$, $\log_a n \in \Theta(\log_b n)$.
4. For $b > a > 0$, $a^n \in o(b^n)$.
5. For all $a > 0$, $a^n \in o(n!)$.

→ $n!$ is **eventually worse** than any exponential complexity function.

1. Big O가 성립하면 반대로 하고 Omega O도 성립
2. Theta O가 성립하면 반대로 하고 Theta O는 성립
3. $a > 1$ and $b > 1$ 일때 $\log_a N \in \Theta(\log_b N)$ 은 성립
4. $b > a > 0$ 일 때 $a^n \in O(b^n)$ 는 성립한다. → 지수의 밑이 더 크면 영향력이 더 크기 때문.
5. $a^n \in O(n!)$ 는 항상 참이다. → 지수는 팩토리얼의 영향력을 따라갈 수 없다.