

하나와영 C 세미나

SORTING

BIG O NOTATION

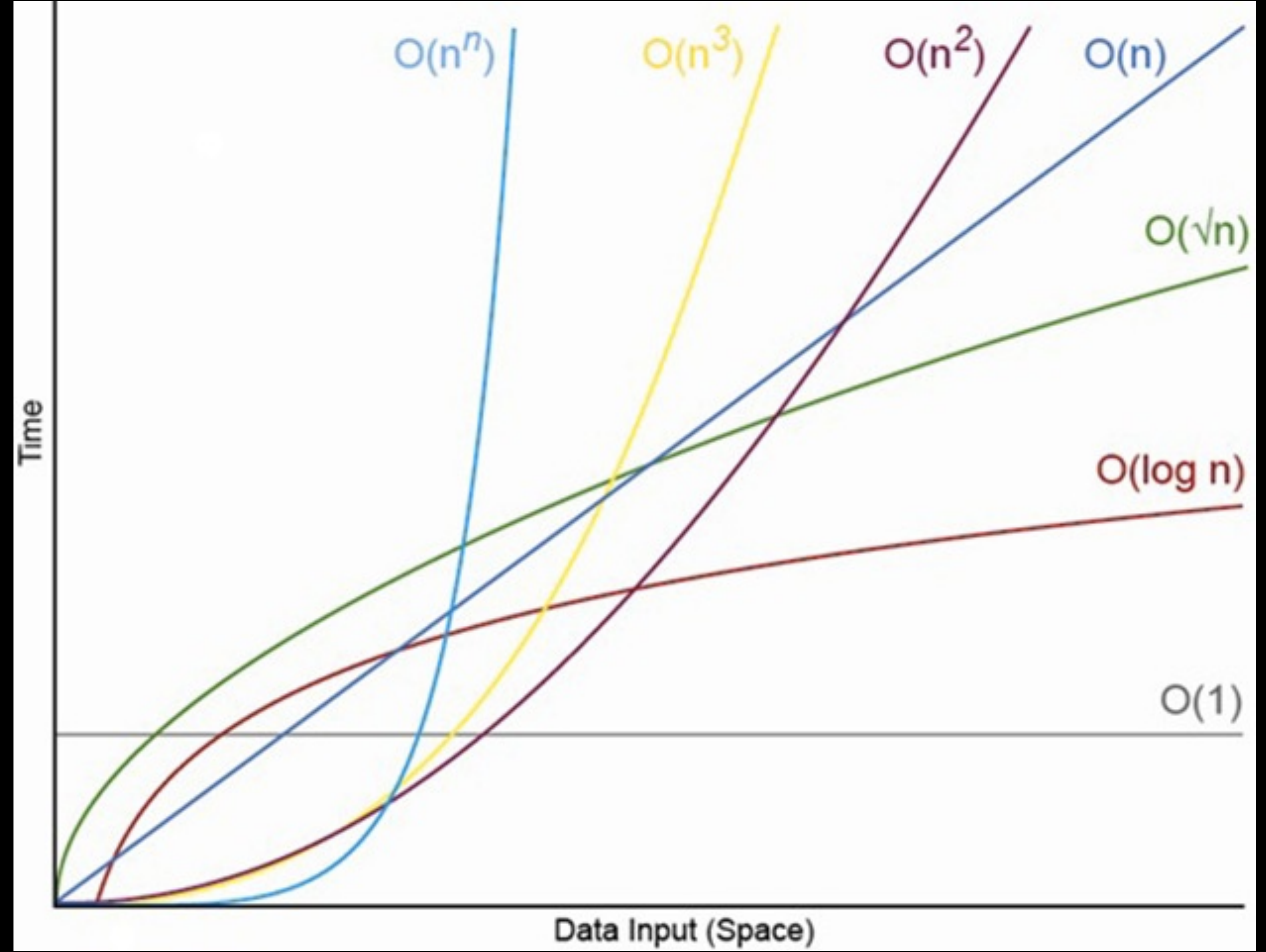
BIG O NOTATION

We worry about the speed of our algorithms for large input sizes.
For large n , $\log(n)/n$, and $n/\exp(n)$ are very small.

However, $n/2^n$ is a constant for all n .

알고리즘의 계산복잡도, 데이터양에 따른 시간 비
알고리즘을 평가할 수 있는 척도이다

e.g, $O(1)$, $O(N)$, $O(N^2)$, $O(\log N)$



정려

PROBLEM

Sort an N-element list of integers in ascending order

Use comparison and swap (assignment) operation

Sort an N-element list of integers in ascending order

Use **comparison** and **swap** (assignment) operation

(**비교**와 **교체**)

일정한 순서가 있는 데이터를 순서대로 분류하는 일

Before Sorting

2	7	1	8	3	5	9	0
---	---	---	---	---	---	---	---

Sorted

0	1	2	3	5	7	8	9
---	---	---	---	---	---	---	---

▶ Bubble ▶ Insert ▶ Select ▶ Shell ▶ Heap ▶ Merge ▶ Quick

BUBBLE SORT ALGORITHM

1. 각 반복마다, 짝을 지어 비교한다. ($x(n)$, $x(n-1)$)
2. 만약 짝이 순서가 맞지 않는다면 교체한다
3. 더 작은 수들이 위로(앞으로) 떠오른다(비눗방울)
4. 처음 iterate할때, 가장 작은 수가 가장 앞으로 온다
5. 1-4를 반복한다

BUBBLE SORT

2	7	1	8	3	5	9	0
---	---	---	---	---	---	---	---

0	2	7	1	8	3	5	9
---	---	---	---	---	---	---	---

0	1	2	7	3	8	5	9
---	---	---	---	---	---	---	---

0	1	2	3	7	5	8	9
---	---	---	---	---	---	---	---

0	1	2	3	5	7	8	9
---	---	---	---	---	---	---	---

BUBBLE SORT

```
for (i=0; i ≤ n-1; i++)  
    for (j=n; j ≥ i+1; j--)  
        if A[j-1] > A[j]  
            then swap A[j-1] and A[j]
```

$$T(n) = \sum (n-i) = n(n-1)/2 = O(n^2)$$

SELECTION SORT ALGORITHM

1. 각 자리에 맞는 원소를 찾는다
2. 그 자리에 있던 원소와 자리를 맞바꾼다

SELECTION SORT

2	7	1	8	3	5	9	0
---	---	---	---	---	---	---	---

0	7	1	8	3	5	9	2
---	---	---	---	---	---	---	---

0	1	7	8	3	5	9	0
---	---	---	---	---	---	---	---

0	1	2	8	3	5	9	7
---	---	---	---	---	---	---	---

0	1	2	3	8	5	9	7
---	---	---	---	---	---	---	---

SELECTION SORT(CONTINUED)

0	1	2	3	8	5	9	7
---	---	---	---	---	---	---	---

0	1	2	3	5	8	9	7
---	---	---	---	---	---	---	---

0	1	2	3	5	7	9	8
---	---	---	---	---	---	---	---

0	1	2	3	5	7	8	9
---	---	---	---	---	---	---	---

SELECTION SORT

$$\sum_{k=1 \text{ to } (n-1)} k = (n-1)(n)/2 = (1/2) n^2 + O(n)$$

INSERTION SORT ALGORITHM

1. 각 반복마다, 작은 원소를 찾는다
2. 각각의 원소마다, 적절한 원소에 채워넣는다.

INSERTION SORT

2	7	1	8	3	5	9	0
---	---	---	---	---	---	---	---

2	7	1	8	3	5	9	0
---	---	---	---	---	---	---	---

1	2	7	8	3	5	9	0
---	---	---	---	---	---	---	---

1	2	7	8	3	5	9	0
---	---	---	---	---	---	---	---

1	2	3	7	8	5	9	0
---	---	---	---	---	---	---	---

INSERTION SORT(CONTINUED)

1	2	3	5	7	8	9	0
---	---	---	---	---	---	---	---

1	2	3	5	7	8	9	0
---	---	---	---	---	---	---	---

0	1	2	3	5	7	8	9
---	---	---	---	---	---	---	---

INSERTION SORT

```
for( i=1;i<=n-1;i++)do
    tmp = A[i];
    for( j = i; j > 0 && tmp < A[j-1]; j--) do
        A[j] = A[j-1]; A[j] = tmp;
```

$$T(n) = \sum(i) = n(n-1)/2 = O(n^2)$$