Data Structures and Algorithms

Lecture 03

Aniket Basu Roy

BITS Pilani Goa Campus

2026-01-16 Mon

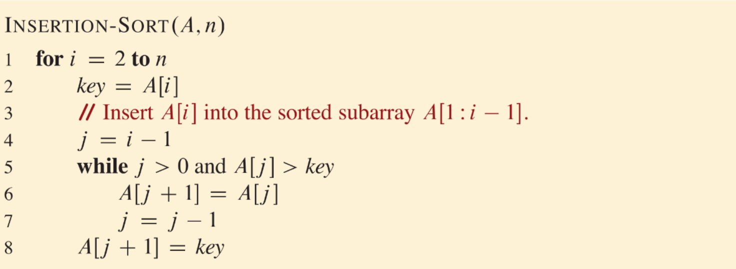
Agenda

Order of Growth

Asymptotic Notation

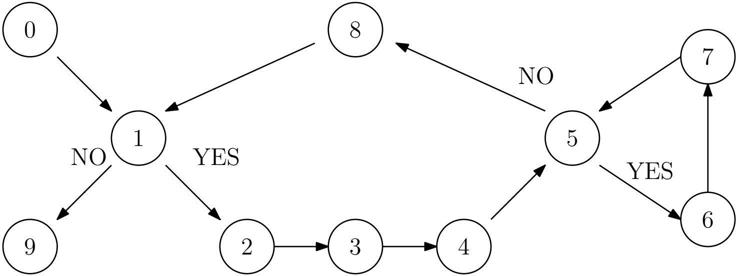
- O-notation
- Ω-notation
- Θ-notation
- o-notation
- ω-notation

Insertion Sort

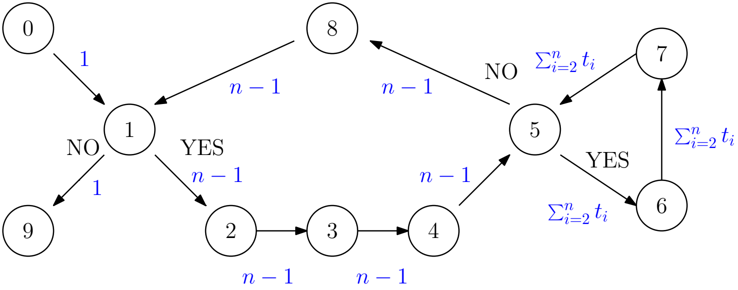


```
INSERTION-SORT($A, n$)
for $i = 2$ to $n$
    $key = A[i]$
    // Insert $A[i]$ into the sorted subarray $A[1 : i - 1]$.
    $j = i - 1$
    while $j > 0$ and $A[j] > key$
        $A[j + 1] = A[j]$
        $j = j - 1$
    $A[j + 1] = key$
```

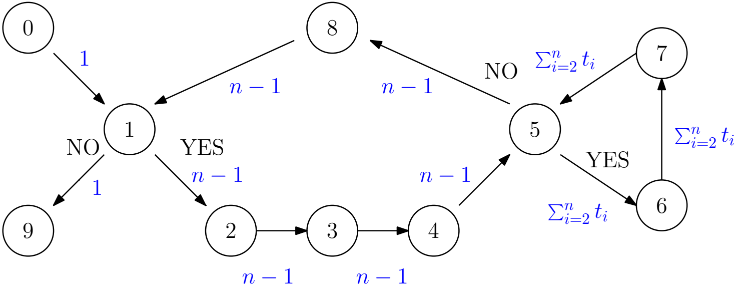
Insertion Sort



Insertion Sort

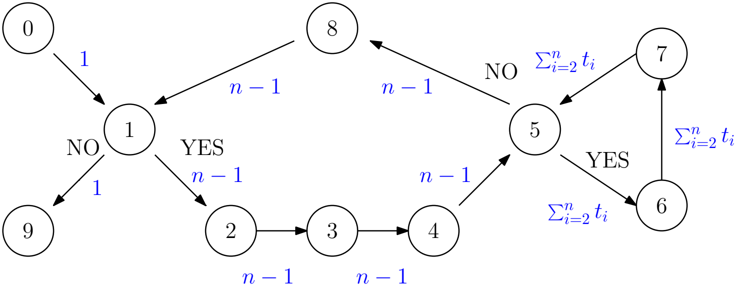


Insertion Sort



c_0	1	c_5	$\sum_{i=2}^{n} t_i + n - 1$
c_1	n	c_6	$\sum_{i=2}^{n} t_i$
c_2	$n-1$	c_7	$\sum_{i=2}^{n} t_i$
c_3	$n-1$	c_8	$n-1$
c_4	$n-1$	c_9	1

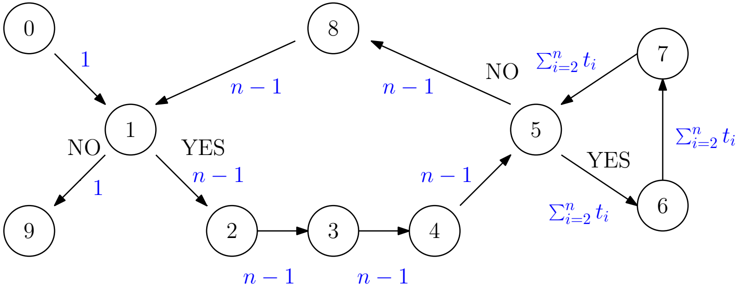
Insertion Sort



$$T(n) = \Big(\sum_{i=2}^{n} t_i\Big) \cdot (c_5 + c_6 + c_7) + n \cdot (c_1 + c_2 + c_3 + c_4 + c_5 + c_8)$$

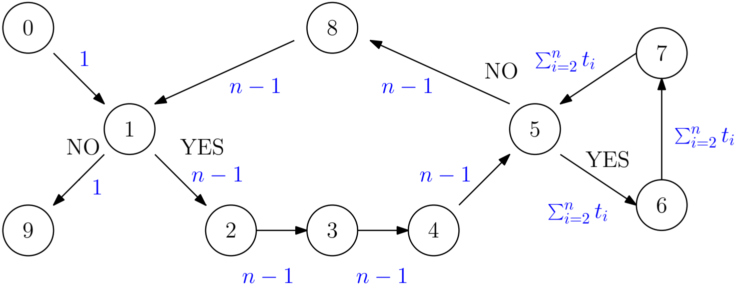
$$+ (c_0 - c_2 - c_3 - c_4 - c_5 - c_8)$$

Insertion Sort



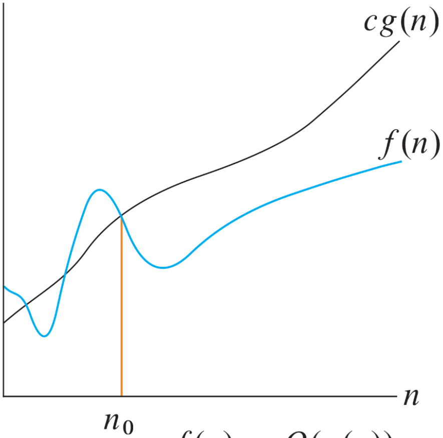
$$T(n) \leq a \cdot n^2 + b \cdot n + c$$

Insertion Sort



$$T(n) = O(n^2)$$

$$f(n) = O(g(n))$$



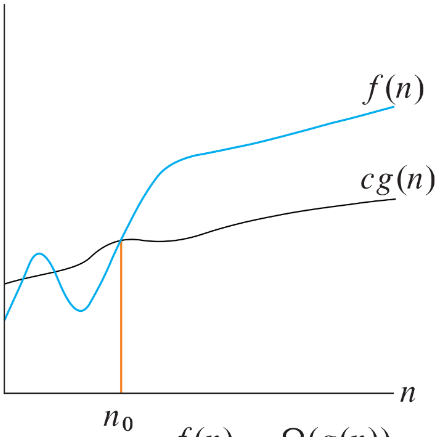
$$f(n) = O(g(n))$$

Formal Definition

$$O(g(n)) = \{f(n) \mid \exists c > 0, n_0 > 0,\ \ \forall n \geq n_0,\ \ 0 \leq f(n) \leq c{\cdot}g(n)\}$$

pronounced as *big-oh of g of n*

$$f(n) = \Omega(g(n))$$



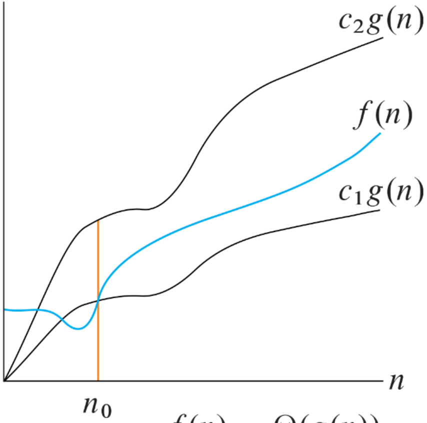
$f(n) = \Omega(g(n))$

Formal Definition

$$\Omega(g(n)) = \{f(n) \mid \exists c > 0, n_0 > 0,\ \ \forall n \geq n_0,\ \ 0 \leq c \cdot g(n) \leq f(n)\}$$

pronounced as *big-omega of g of n*

$$f(n) = \Theta(g(n))$$



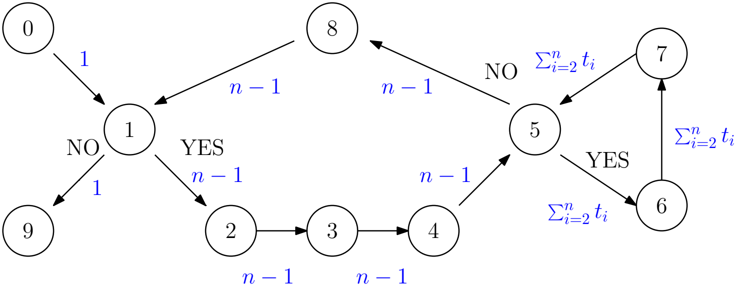
$$f(n) = \Theta(g(n))$$

Formal Definition

$$\Theta(g(n)) = \{f(n) \mid \exists c_1, c_2, n_0 > 0, \;\; \forall n \geq n_0,$$
$$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)\}$$

pronounced as *theta of g of n*

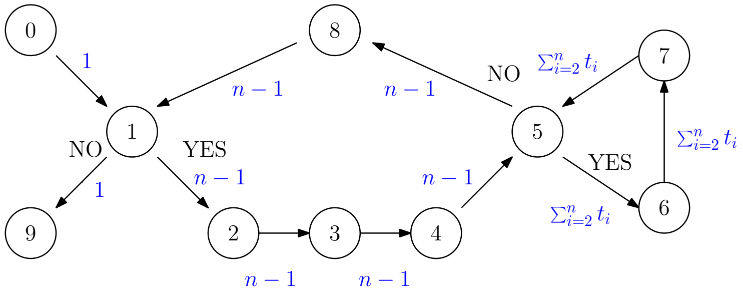
Insertion Sort



$$T(n) = \left(\sum_{i=2}^{n} t_i\right) \cdot (c_5 + c_6 + c_7) + n \cdot (c_1 + c_2 + c_3 + c_4 + c_5 + c_8)$$

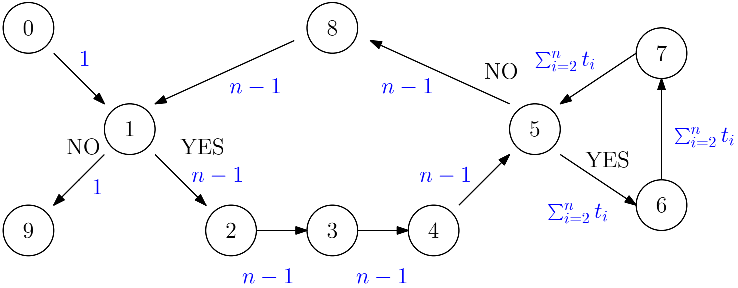
$$+ (c_0 - c_2 - c_3 - c_4 - c_5 - c_8)$$

Insertion Sort



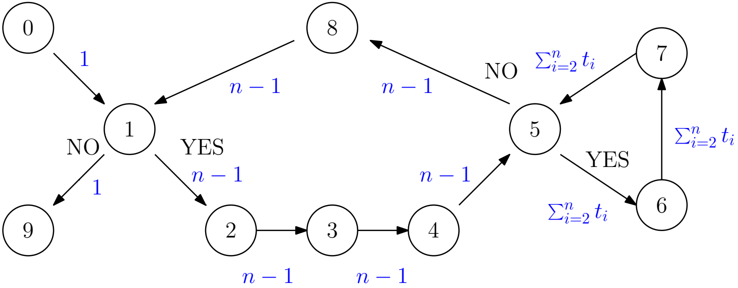
$$T(n) = \left(\sum_{i=2}^{n} t_i \right) \Theta(1) + n \cdot \Theta(1) + \Theta(1)$$

Insertion Sort



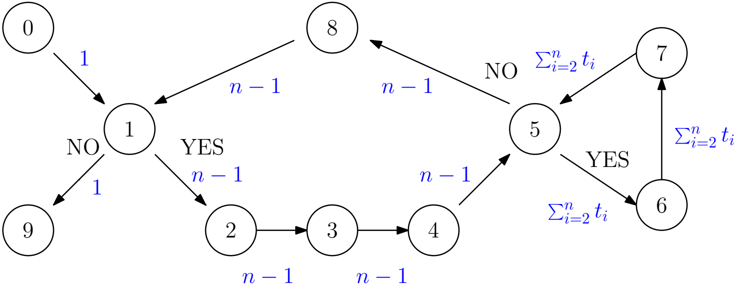
$$T(n) = \Big(\sum_{i=2}^{n} t_i\Big)\Theta(1) + \Theta(n) + \Theta(1)$$

Insertion Sort



$$T(n) = O(n^2)\Theta(1) + \Theta(n) + \Theta(1)$$

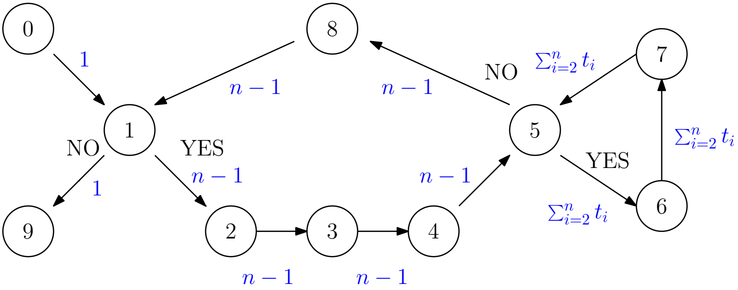
Insertion Sort



Worst-case time complexity:

$$T(n) = \Omega(n^2)\Theta(1) + \Theta(n) + \Theta(1)$$

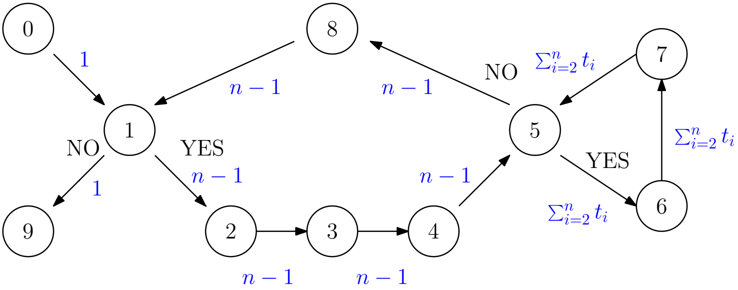
Insertion Sort



Worst-case time complexity:

$$T(n) = \Theta(n^2)\Theta(1) + \Theta(n) + \Theta(1)$$

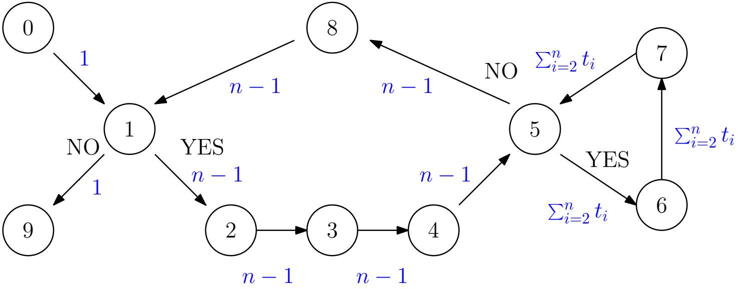
Insertion Sort



Worst-case time complexity:

$$T(n) = \Theta(n^2)$$

Insertion Sort



Best-case time complexity:

$$T(n) = \Theta(n)$$

$$f(n) = o(g(n))$$

Formal Definition

$$o(g(n)) = \{f(n) \mid \forall c > 0,\ \exists n_0 > 0,\ \forall n \geq n_0,\ \ 0 \leq f(n) < c \cdot g(n)\}$$

pronounced as *small-oh of g of n*

Formal Definition

$$o(g(n)) = \{f(n) \mid \forall c > 0,\ \exists n_0 > 0,\ \forall n \geq n_0,\ \ 0 \leq f(n) < c{\cdot}g(n)\}$$

pronounced as *small-oh of g of n*

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

$$f(n) = \omega(g(n))$$

Formal Definition

$$\omega(g(n)) = \{f(n) \mid \forall c > 0,\ \exists n_0 > 0,\ \forall n \geq n_0,\ \ 0 \leq c{\cdot}g(n) < f(n)\}$$

pronounced as *small-omega of g of n*

Formal Definition

$$\omega(g(n)) = \{f(n) \mid \forall c > 0,\ \exists n_0 > 0,\ \forall n \geq n_0,\ \ 0 \leq c{\cdot}g(n) < f(n)\}$$

pronounced as *small-omega of g of n*

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$