INSERTION SORT

Γ	9	7	6	15	17	5	10	11
L				10	.,		10	•••
. [9	7	6	15	17	5	10	11
1	7	9	6	15	17	5	10	11
	6	7	9	15	17	5	10	11
I	_	I _		T			l	.
	6	/	9	15	1/	5	10	11
	6	7	9	15	17	5	10	11
1	U	- 1	9	13	17	بر	10	11
`						_		
	5	6	7	9	15	17	10	11
					N		ノ	
	5	6	7	9	10	15	17	11
						*		
	5	6	7	9	10	11	15	17

INSERTION SORT

INSERTION-SORT (A)		cost	times
1	for $j = 2$ to A.length	c_1	n
2	key = A[j]	c_2	n-1
3	// Insert $A[j]$ into the sorted		
	sequence $A[1j-1]$.	0	n-1
4	i = j - 1	C4	n-1
5	while $i > 0$ and $A[i] > key$	c_5	$\sum_{j=2}^{n} t_j$
6	A[i+1] = A[i]	c_6	$\sum_{j=2}^{n} (t_j - 1)$
7	i = i - 1	c_7	$\sum_{j=2}^{n} (t_j - 1)$
8	A[i+1] = key	c_8	n-1

Example Recurrences for Algorithms

· Insertion sort

$$T(n) = \begin{cases} 1 & \text{for } n \le 1 \\ T(n-1) + n & \text{otherwise} \end{cases}$$

· Linear search of a list

$$T(n) = \begin{cases} 1 & \text{for } n \le 1 \\ T(n-1)+1 & \text{otherwise} \end{cases}$$

INSERTION SORT

HOMEWORK

Common Recurrence Relations

Recurrence Relation	Result	Example
T(n) = T(n/2) + O(1)	$T(n) = O(\log n)$	Binary search, Euclid's GCD
T(n) = T(n-1) + O(1)	T(n) = O(n)	Linear search
T(n) = 2T(n/2) + O(1)	T(n) = O(n)	
T(n) = 2T(n/2) + O(n)	$T(n) = O(n\log n)$	Merge sort (Chapter 24)
$T(n) = 2T(n/2) + O(n\log n)$	$T(n) = O(n\log^2 n)$	
T(n) = T(n-1) + O(n)	$T(n) = O(n^2)$	Selection sort, insertion sort
T(n) = 2T(n-1) + O(1)	$T(n) = O(2^n)$	Towers of Hanoi
T(n) = T(n-1) + T(n-2) + O(1)	$T(n) = O(2^n)$	Recursive Fibonacci algorithm