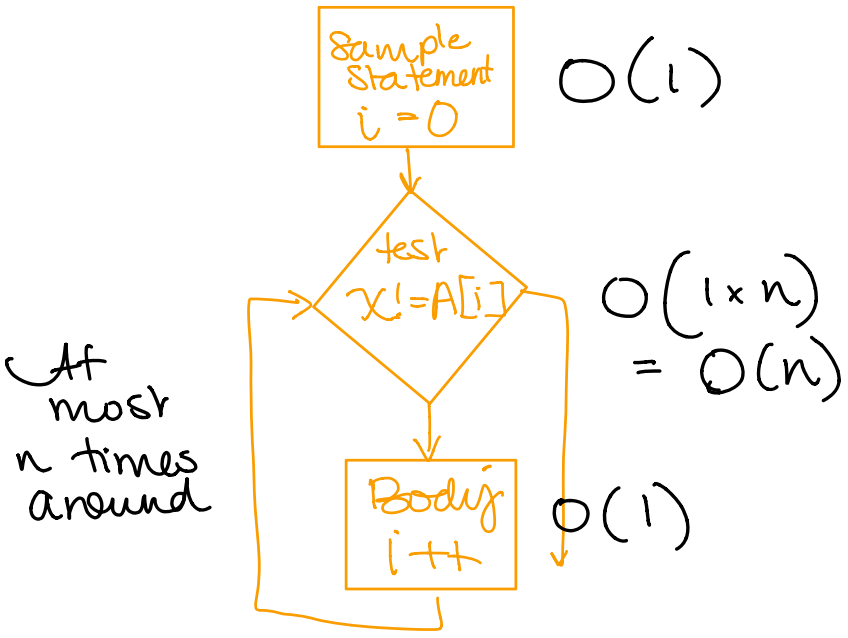
Lecture 24

Ex. A linear search or sequential search is a method for finding an element w/in a list n

This program searches an array $A[0 \ldots n-1]$ to find element x believed to be in the array

```
(1) i=0
(2) while (x!=A[i])
(3)       i++
```



The running time of the entire program fragment is $O(n)$

Note: This $O(n)$ program can be replaced by $O(\log n)$ prog. using the so-called

Binary Search

Sorting Algorithm - puts elements in list in a certain order

Ex. Selection sort is a sorting algorithm (in-place sort)

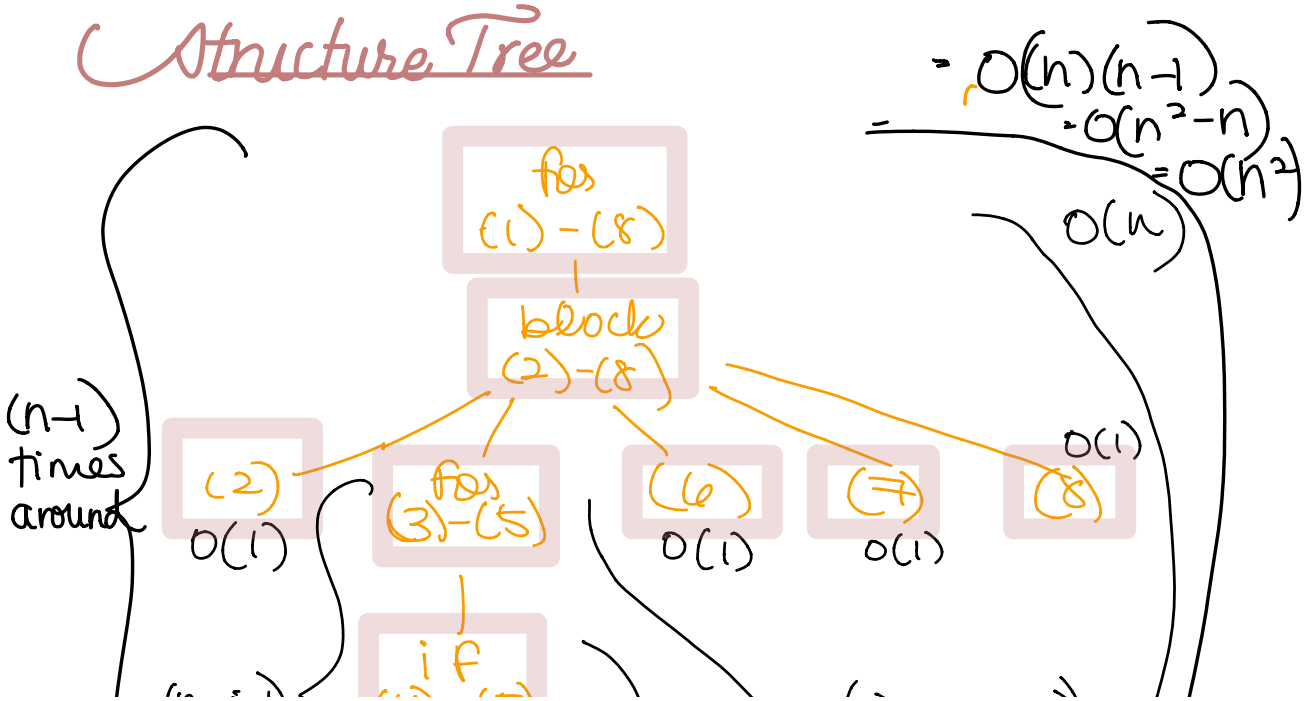
UNSORTED SUBLIST	SMALLEST ELEMENT IN LIST	SORTED SUBLIST
(11, 25, 12, 22, 64)	11	( )
(25, 12, 22, 64	12	(11)
(25, 22, 64)	22	(11, 12)
(25, 64)	25	(11, 22)
(64)	64	(11, 22, 25)
( )		(11, 22, 25, 64)

Selection-sort prog. fragment

```
(1) for (i=0; i<n-1, i++)
(2)     small=i
(3)     for (j=i+1; j<n; j++)
(4)         if (A[j] < A[small])
(5)             small=j;
(6)     swap = A[small];
(7)     A[small] = A[i];
(8)     A[i] = swap
```

Running time = quadratic (n^2) — two loops

Structure Tree

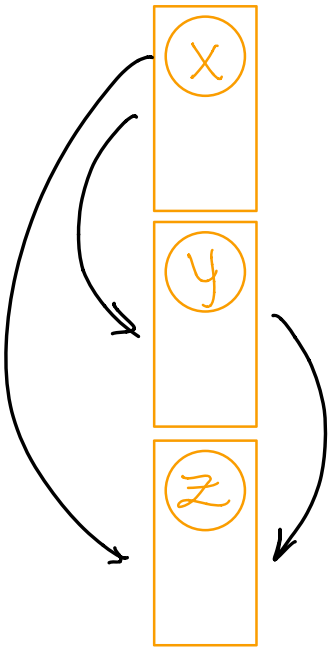


(n-i-1) times around ((4)-(5) ; (5) O(1)) O(1)

O(1) (n-i-1)
= O(n-i-1)
= O(n)

Conclusions:

1) Linear search has linear complexity
2) Selection-sort has quadratic complexity



Linear-search - n
selection-sort - n^2
Binary search - log n

Analyzing Programs w/ Function Calls

Recursive Function - calls itself

Nonrecursive - does NOT call itself

1. Evaluate running times of functions that do NOT call any other function
2. Evaluate running times of ONLY functions whose running time we've already determined
3. Repeat

Homework:

https://u.osu.edu/alzalg.1/files/2019/10/hw10.pdf