Lecture 1

Monday, January 11, 2021 1:38 PM

- Space Complexity : $S(n)$
- Time Complexity : $T(n)$

<u>Sorting Problem</u>

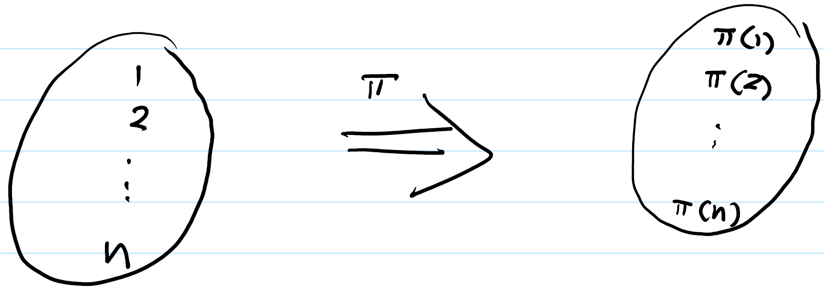
Input : $A = [a_1, a_2, \dots, a_n]$

$size(A) = n$

<u>output</u> : $A' = [a_{\pi(1)}, a_{\pi(2)}, \dots, a_{\pi(n)}]$

such that, s.t.

min(A) $\rightarrow a_{\pi(1)} < a_{\pi(2)} < a_{\pi(3)} \vdots < a_{\pi(n)}$ max(A)



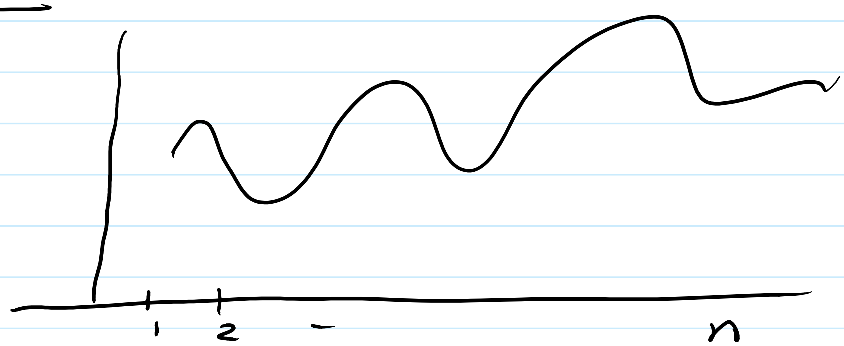
π : is a permutation of the indices $(1, 2, 3, \dots, n)$

Look-up-table LUT

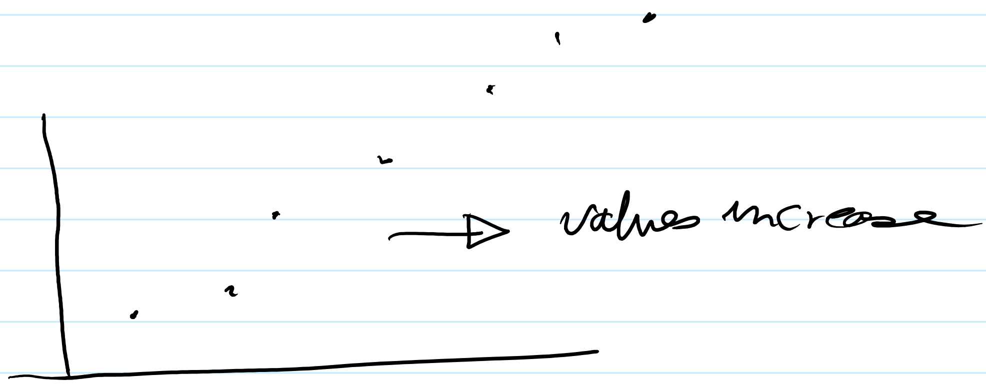
i	$\pi(i)$
1	27
2	63
3	7
100	29

plot A

Plot A



Plot A'



Method #1 'Naive Design'

Recall # of permutation of n-objects

$= n!$

n	$n!$
2	2
3	6
4	24
⋮	
$10!$	$10\cdot9\cdot8\cdot7\cdot6\cdot3\cdot2\cdot1$

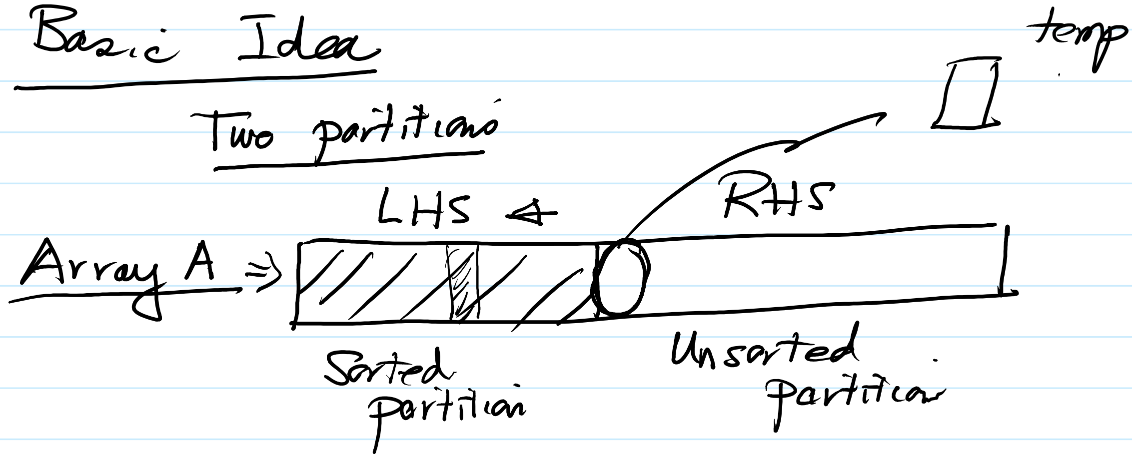
Method #2: Insertion Sort

It is an iterative method

It is an iterative method

Basic Idea

Two partitions



Algorithm (IS)

```
Insertion_Sort (A)
   = size(A)
  for j = 2 to n
      temp = A(j)
      i = j-1
      while (i>0 and A(i) > temp)
            A(i+1) = A(i)
            i = i-1
      A(i+1) = temp
```

Numerical Example

							temp
5	2	4	6	1	3		2
2	5	4	6	1	3		4
2	4	5	6	1	3		6
2	4	5	6	1	3		1
1	2	4	5	6	3		3

1	2	3	4	5	6

Analysis of Insertion Sort

(1) Correctness

What is the Loop invariant property?

LHS partition is sorted

(a) Initialization of LHS

LHS has one element, thus Loop invariant exits

(b) Maintenance

Sorting property of LHS holds after inserting the value in temp

(c) termination

When $j = n+1$, the for Loop is terminated and LHS is the sorted array.

$\boxed{3}$ Space Complexity?

→ temp, j, i

S

space

$S(n) = \underline{n+3} \rightarrow$ temp, j, i

$\boxed{\text{Read chapters 1 and 2}}$

<u>Matlab</u>