

L6

Sorting Problem

Input : A sequence of n numbers $\langle a_1, a_2, a_3, a_4, \dots, a_n \rangle$

Output: A permutation (reordering) of $\langle a_1', a_2', a_3', a_4', \dots, a_n' \rangle$ such that $a_1' \leq a_2' \leq a_3' \leq a_4' \leq \dots \leq a_n'$

• An algorithm for sorting : Insertion sort

• Works in the same way as many people sort a hand of playing cards

- Start with an empty left hand and all cards face down on the table
- Remove one card at a time from the table and insert it to the correct position in the left hand

Design : Insertion sort

- Input Cards: 5 , 7, 2, 3, 7, 10
- Input cards : 5 Hearts , 7 Clubs , 2 Spades, 3 Diamonds, 7 Hearts, 10 Clubs
- Output : 2 Spades, 3 Diamonds, 5 Hearts, 7 Hearts, 7 Clubs, 10 Clubs ???
- What all are the steps we did?
- We took the first card, we just placed it in the left hand
- We took the second card, we compared it with the first card in the hand and placed it in the proper position
- When we took the third card, did we compare it with the first card and then the second card (in that order)?

Observations

- When we take any card (say *key card*), we start comparing it with the last card in our left hand
- If the value of the *key card* is less than the last card in our left hand, then only we proceed with further comparisons
- If the value of the *key card* is equal to or greater than the last card in our left hand, then we place the key card as the last card
- This action is performed based on an observation
- Cards in the left hand is already sorted
- At any time, we are inserting to a sorted list of elements

How do we write the algorithm?

- What is the input of our algorithm?
- Input: array of elements 5, 7, 2, 3, 7, 10 . We pass the input array to the function insertion sort
- Keep 5 as such in the first position of the array, assuming that it is sorted by itself
- Take 7, compare it with 5, place it there itself
- Take 2, we put in a temporary variable (say *key*), we shift 7 to 2's position in the array, now the array will be 5,7,7,3,7,10
- Do we have to place *key* (2) in the first 7's position and make the array as 5,2,7,3,7,10 ?

Design of insertion sort -contd.

- Input: array of elements 5, 7, 2, 3, 7, 10 .
- Compare key with 5 and make the array 5,5,7,3,7,10
- How many loops we should have?
- One loop for sure which goes from 1 to n
- Another loop which goes from position of *key* element to 1 ?

Insertion Sort

INSERTION SORT(A)

1. for $j=2$ to $A.length$
2. $key = A[j]$;
3. //Insert $A[j]$ into the sorted sequence $A[1...j-1]$
4. $i = j-1$
5. while $i > 0$ and $A[i] > key$
6. $A[i+1] = A[i]$
7. $i = i-1$
8. $A[i+1] = key$

Trace INSERTION SORT(A)

- Input : 5, 2, 4, 6, 1, 3
- First iteration of *for* loop , with $j=2$
- Key =2, $i=1$
- *while* loop is executed as $i>0$ and $A[i] > \text{key}$
- $A[2] = 5$, $i=0$
- *while* loop fails to execute as $i=0$
- $A[0+1] = 2$
- Intermediate output after the 1st iteration of *for* loop : 2, 5, 4, 6, 1, 3

INSERTION SORT(A)

1. for $j=2$ to $A.\text{length}$
2. $\text{key} = A[j];$
3. //Insert $A[j]$ into the sorted sequence $A[1...j-1]$
4. $i = j-1$
5. while $i > 0$ and $A[i] > \text{key}$
6. $A[i+1] = A[i]$
7. $i = i-1$
8. $A[i+1] = \text{key}$

Insertion sort

- Insertion sort is an **in place** sorting method ie it rearranges the numbers within the input array, with at most a constant number of them stored outside the array at any time.
- The input array A contains the sorted sequence after the procedure is finished
- **Operations of insertion sort on an array A = 1, 4, -2, -3**
- After 1st iteration of *while* loop, with $j = 2$, $i = 1$: A = 1, 4, -2, -3
- After 1st iteration of *for* loop, with $j = 2$: A = 1, 4, -2, -3

INSERTION SORT(A)

1. for $j=2$ to A.length
2. key = A[j];
3. //Insert A[j] into the sorted sequence A[1...j-1]
4. i = j-1
5. while $i > 0$ and A[i] > key
6. A[i+1]=A[i]
7. i=i-1
8. A[i+1]= key

Operation of Insertion Sort on A

- **A = 1, 4, -2, -3**
- After 1st iteration of *while* loop, with $j = 3, i = 2$: A = 1, 4, 4, -3
- After 2nd iteration of *while* loop, with $j = 3, i = 1$: A = 1, 1, 4, -3
- After 2nd iteration of *for* loop, with $j = 3$: A = -2, 1, 4, -3
- After 1st iteration of *while* loop, with $j = 4, i = 3$: A = -2, 1, 4, 4
- After 2nd iteration of *while* loop, with $j = 4, i = 2$: A = -2, 1, 1, 4
- After 3rd iteration of *while* loop, with $j = 4, i = 1$: A = -2, -2, 1, 4
- After 3rd iteration of *for* loop, with $j = 4$: A = -3, -2, 1, 4

INSERTION SORT(A)

1. for $j=2$ to A.length
2. key = A[j];
3. //Insert A[j] into the sorted sequence A[1...j-1]
4. i = j-1
5. while $i > 0$ and A[i] > key
6. A[i+1] = A[i]
7. i = i-1
8. A[i+1] = key

Thank You