## Danyl Fernandes 2020012004 (72) 11-04-2021

## **AOA Experiment 1**

## Aim:

To implement & analyze Insertion Sort Algorithm and compare its complexity with Selection Sort:

## Implementation:

```
• • •
 1 // Authored by: Danyl Fernandes
 3 #include <bits/stdc++.h>
 4 using namespace std;
 6 void sort(int arr[], int n) {
       int key = 0;
 8
 9
       for (int i = 1; i < n; i++) {
10
           key = i;
11
           for (int j = key-1; j \ge 0; j--) {
12
                if (arr[key] < arr[j]) {</pre>
13
                    int temp = arr[j];
14
15
                    arr[j] = arr[key];
16
                    arr[key] = temp;
17
                    key = j;
18
19
           }
20
       }
21
22
       for (int i = 0; i < n; i++) {
           cout << arr[i] << " ";
23
24
       }
25 }
26
27 int main() {
       int arr[] = {4, 2, 1, 6, 9, 13, 3, 8, 5};
28
29
       sort(arr, 9);
30 }
31
```

```
1 2 3 4 5 6 8 9 13
Process returned 0 (0x0) execution time : 0.016 s
Press any key to continue.
```

•	Danyl Fernandes
	2020012004 (72)
	Exp 01
	Tham
	Therry
	- Insertion sort uses the analogy of sorting
	a time from the deck 4 it is inserted a
	a time from the deck 4 it is insuted a
	the correct location in thand itself.
	Upcoming couds are processed in the san
	way
_	To insert a new cord, all the cards in
1	hand have solve layer than now and
	hand having value larger than new could
	are shifted on the right side by one.
	New cold is insuted on space created after moving some 'k' colds on the right
	after moring some 'K' couds on the right
	side.
Andrew	1 My copy of a special fall of the first of
	Insertion sort is an inplace algorithm.
	It does not sequie extra memory. Sortin
	is done in the linket were itself . In
	theretion it I first it classest up
	Marion K That K extrems are
	is done in the input away itself. In iteration 'k' first 'k' elements are always sorted.
2 /	
-	Ranning time is the number of steps
	required to solve the problem on RAM
	model. Each instruction may take differ
As parameters and the second s	lequied to solve the peoplem on RAM model. Each instruction may take diffeer amount of time.
	U .

	Dod Franks
· ·	Danyl Fernandes
	2020012004 (72)
	Analysis:
	From the algorithm below, analysis can be done:
	$1 - (1+\alpha)\alpha = 1 \leq 1$
a a	1 fox j = 2 to length [A]
	2 do key \ ALj]
	in court AliT into some
	3 insert A[j] into solled seq.
	4 1 (-1)
	while i>0 and A[i]> key  do A[i+1] ← A[i]
-	7 i ← i-1
	8 A[i+1] Key
	late the land of the
-	we can compute the time by suming up the
	We can compute the time by suming up the time taken by each instruction
-112,	
*	$I(n) = c_1 n + c_2(n-1) + c_3(n-1)$
	$T(n) = c, n + c_2(n-1) + c_3(n-1) + c_4(z^{n^2}+j) + c_5 z^{n} (t^{n-1})$
-	0
	+ ( ( = tj-1) + cz (n-1)
	Best case
	Happens when data is already sorted.
	T(a) = a + a + b + a + b + a + b + a + b + a + b + a + b + a + b + a + b + a + b + a + b + a + b + a + b + a + b + a + b + a + a
-	$T(n) = c_1 n + c_2(n-1) + c_3(n-1) \cdot c_4(n-1)$
1	$= (C_1 + C_2 + C_3 + C_n + C_7)n$
	$-(c_1+c_2+c_3+c_n+c_7)$
	= lineal function of $n$ T(n) = O(n)
	1(n) = 0(n)
-	( a) C mand i trous i gas i tisdi
-	The side side side side side side side sid

Danyl Fernandus 2020012004 (72) Worst case Data sated in descending order Hence, tj=j for j=2,3....  $\sum_{i=2}^{n} j = \frac{n(n+1)-1}{2}$  $\sum_{i=2}^{n} (j-i) = n(n-i)$  $C_{n} = C_{n} + C_{2}(n-1) + C_{3}(n-1)$   $C_{n} \left[ \frac{n(n+1)-1}{2} + C_{5} \left[ \frac{n(n-1)}{2} \right] \right]$ + C [n(n-1)] + Cq (n-1)  $= \left(\frac{n+c_5+c_6}{2}\right)n^2 + \left(\frac{c_1+c_2+c_3}{2}+\frac{c_1+c_2+c_3}{2}\right)n^2$ - (C2 + C3 + Cn + Cq) = quadratic function of n  $T(n) = O(n^2)$ Compaison with selection soit: - Best case of insertion soft is O(n) A its worst case is  $O(n^2)$ Where as for selection soft, both best case & worst have  $O(n^2)$ complexity

- Hence, we concur that insertion soit electron fort.  Example:  a) 5 2 4 6 1 3  b) 2 5 4 6 1 3  c) 2 4 5 6 1 3  e) 1 2 4 5 6 3  f) 1 2 3 4 5 6  Conclusion: We implemented the insersort, analyzed it & successfully composits complexity with that of electron	- 11	200	1. 0			1.61			1
Example:  a) 5 2 4 6 1 3  b) 2 5 4 6 1 3  c) 2 4 5 6 1 3  d) 2 4 5 6 1 3  e) 1 2 4 5 6 3  f) 1 2 3 4 5 6	lelat	ull	, we	u H	han	hout Sele	etion	estor Pos	t.
a) 5 2 4 6 1 3 b) 2 5 4 6 1 3 c) 2 4 5 6 1 3 d) 2 4 5 6 1 3 e) 1 2 4 5 6 3 f) 1 2 3 4 5 6			V						
b) 2 5 4 6 1 3 c) 2 4 5 6 1 3 d) 2 4 5 6 1 3 e) 1 2 4 5 6 3 f) 1 2 3 4 5 6	Exam	b/e.						-	•
c) 2 4 5 6 1 3  d) 2 4 5 6 1 3  e) 1 2 4 5 6 3  f) 1 2 3 4 5 6	a)	5	2	4	6	1	3		
c) 2 4 5 6 1 3  d) 2 4 5 6 1 3  e) 1 2 4 5 6 3  f) 1 2 3 4 5 6	b)	2	5	4	6	1	3	9	
4) 2 4 5 6 1 3 e) 1 2 4 5 6 3 f) 1 2 3 4 5 6									
e) 1 2 4 5 6 3 f) 1 2 3 4 5 6	c)	2	4	5	6	1	3	-	
f) 1 2 3 4 5 6	d)	2	4	5	6	1	3		
f) 1 2 3 4 5 6	(0	- 1	2	1.	5	(	2		
	(2)	- 1		7	5		5		
Condusion: We implemented the inser sort, analyzed it & successfully composits complexely with that of Pelectron	f)	1	2	3	4	5	6		
Condusion: We implemented the inser sort, analyzed it & successfully composits complexely with that of delection					8				
sort, analyzed it & successfully composits complexely with that of delection	Const	usion	: V	Ve i	noen	nen1e	d.	the	105-1
its complexity with that of delection	sort	, ar	alyze	d'It	2	Suc	cessf	ely	compo
	its	com	blexul	ty n	ith	that	- of	Pel	'ection
								3	
				7	17.12		*		
					(2)				
			-						