Name- Gurkirat Singh Arora
Date-17th June

Running- Time Analysis

The total time taken for this algorithm will be the Cost of each operation * No. of times its executed.
In a static array for n element, pushing 1 element in the stack requires creating a new static array,
pushing the new element at index 0 and copying the rest of the elements from index 1. The time complexity will be O(n)
for pushing single element to the Stack.
Pushing 1 element to an empty Stack, number of possible operation is only 1.
Pushing 1 elements to a Stack when it already has one element, total number of possible operations are 2.(approx)
Pushing 1 elements to a Stack when it already has two element, number of possible operations are 3. (approx)
.
.
.
Pushing 1 elements to a Stack when it already has n element, number of possible operations are n.(approx)
Similarly, for pushing 1 elements to the Stack when It already has n elements,
number of possible operations will be n. (approx)
Therefore, total running time required for pushing n- elements to the stack will be
1+2+3+.....n =n(n+1)/2= n^2+n/2 =n^2(Dominant term when n is very large)
The time complexity for pushing n-elements to the stack wil be O(n^2).

Similarly, When the number of elements in the static array are n ,to pop the element at index 0
requires to shift the elements from index i=1 till i=n-1 , one index towards left.
Therefore, number of operations required to pop 1 element will be n.The time complexity for this operation will O(n).
Popping 1 element from the Stack having n elements, number of possible operations are n.(approx)
Popping 1 elements from the Stack having n-1 elements, total number of possible operations are n-1.(approx)
Popping 1 elements from the Stack having n-2 elements, number of possible operations are n-2. (approx)
.
.
.
Popping 1 elements from the Stack having 1 elements, number of possible operations is 1. (approx)
Therefore, the number of operations required to pop out all the elements from the stack (implemented using static array of
size n) will be n + (n-1) + (n-2) + (n-3) +.......1 = n(n+1)/2 = n^2+n/2 =n^2(Dominant term when n is very large).
The time complexity for popping out n-elements from the Stack will be O(n^2).