



AN ALGORITHM FOR SSD

Salma Sadiqha

Department of Computer Science and Engineering
Yanbu University College, Royal Commission at Yanbu, Yanbu Al-Sinaiyah, Kingdom of Saudi Arabia.

ABSTRACT

In this paper, we present “An algorithm for SSD” which emulates the digits in a seven segment display form. The SSD algorithm is a flexible approach uses an object oriented. The SSD algorithm uses stacks and recursive calls used to display digits. The efficiency will be discussed by comparing “The SSD algorithm” to a digital display with stack and without using stacks.

Keywords: Stacks, application, recursion, object oriented.

INTRODUCTION

In this paper we will compare the features of the object oriented and procedural based language. In this research we will test only the two applications “The SSD algorithm” to other application displaying digits without using stacks.

The procedure oriented programming focuses on how to accomplish a task, the order of the instructions is important. Disadvantage of the procedural oriented language is that user views problem as series of steps rather than objects. The problems stated in objects are better visualized.

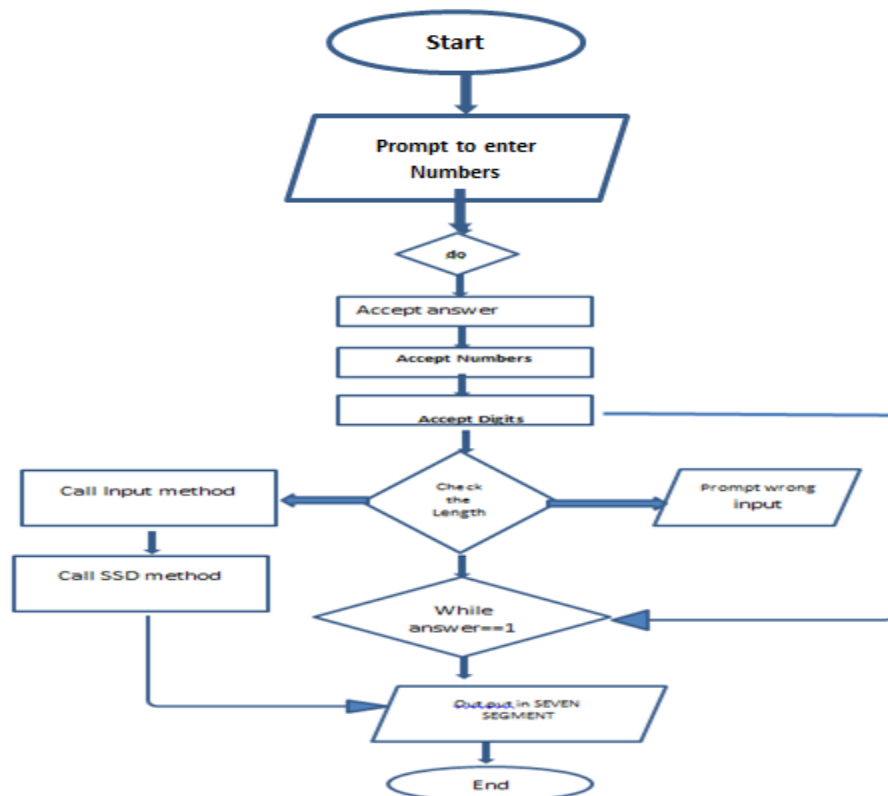
Thus an object oriented programming which breaks program as a concept of objects and methods that interacts to perform a specific task. Using object oriented approach the users can visualize

the solutions to problem more easily. Object oriented programs increases programmer's efficiency where users uses encapsulation and reusability concepts.

This paper also uses one of the data structure concept called as stacks. A stack is a container of objects that are inserted and removed according to the last-in first-out (LIFO) principle. In the pushdown stacks only two operations are allowed: pushing the item into the stack, and popping the item out of the stack. A stack is a limited access data structure that elements can be added and removed from the stack only at one end the top. Push operation adds an item to the top of the stack and pop operation removes the item from the top. If we talk about the daily life example like a stack of books; you can remove only the top book, also you can add a new book on the top.

METHODOLOGY

In this research we used object oriented approach which uses stacks. The stack stores digits and pops in reverse order. Input method is recursively called to make all the accepted digits to in a seven segment display format.



AlgorithmMainMethod:

```
input(numbersOfDigits);
do {
Accepts Digits;

If Check(lengthtoDigitLength)
Call Input methods recursively(
Call SSD methods
}
else{
print"ERROR Wrong number of digits ...";
}
Print "to continue press 1? "
Input answer;
}
while (answer==1);
print "DONE!";
}}
```

Algorithm for Input Method:

```
Input(parameters)
use logic to get last digit
push digit into stack
```

Algorithm for Convert to Seven Segment Method:

```
public void SSD(digit){
int ssd = 0;
for (int j=0;j<dig;j++){
if (!stack.isEmpty())
ssd =stack.pop();
check (conv equals ZERO)
```

Call method Zero()

Continue calling all digits.

Comparison Analysis:

	Non Procedure	SSD Algorithm
Input	Time taken in millisecc	Time Taken in Millisecc
10 digits	0	0
20 digits	1	2
30 digits	1	2
30 digits	2	4

Conclusion

The two main objectives of this research are, first to build an object oriented based application for solving real world problems using stacks which provides programmer's efficiency and second is comparing performance of two algorithms "The SSD algorithm" with stacks and without stacks. Finally we can conclude that Object oriented programming is better for improving programmer's efficiency and stacks can be used as a programmer's tool for various real world problems solving with better performance.

REFERENCES

[1] <http://iacis.org/iis/2004/Chen.pdf>

[2] OBJECT ORINETED PROGRAMMING TEXT BOOK

[3] Data Structures and Algorithms in Java (2nd Edition) 2nd Edition by Robert Lafore.

