Sorting algorithm visualization : Insertion Sort

Last Updated : 02 Nov, 2023

An algorithm like Insertion Sort can be understood easily by visualizing. In this article, a program that visualizes the Insertion Sort Algorithm has been implemented.

The Graphical User Interface(GUI) is implemented in python using pygame library.

Approach:

- Generate random array and fill the pygame window with bars. Bars are straight vertical lines, which represents array elements.
- Set all bars to green color (unsorted).
- Use **pygame.time.delay()** to slow down the algorithm, so that we can see the sorting process.
- Implement a timer to see how the algorithm performs.
- The actions are performed using ‘pygame.event.get()’ method, which stores all the events which user performs, such as start, reset.
- Blue color is used to highlight bars that are involved in sorting at a particular time.
- Orange color highlights the bars sorted.

Observations:

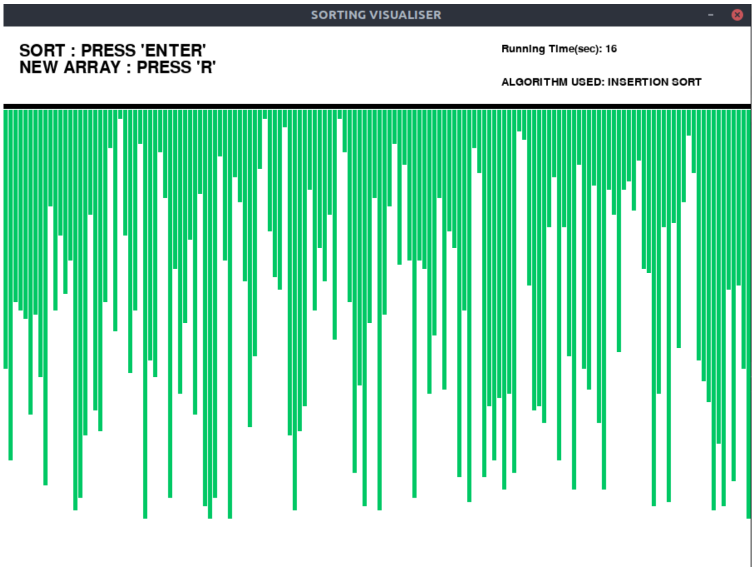
We can clearly see from the Insertion Sort visualization, that insertion Sort is very slow compared to other sorting algorithms like Mergesort or Quicksort.

Examples:

> ***Input:***
> *Press “Enter” key to Perform Visualization.*
> *Press “R” key to generate new array.*

Output:

Initial:

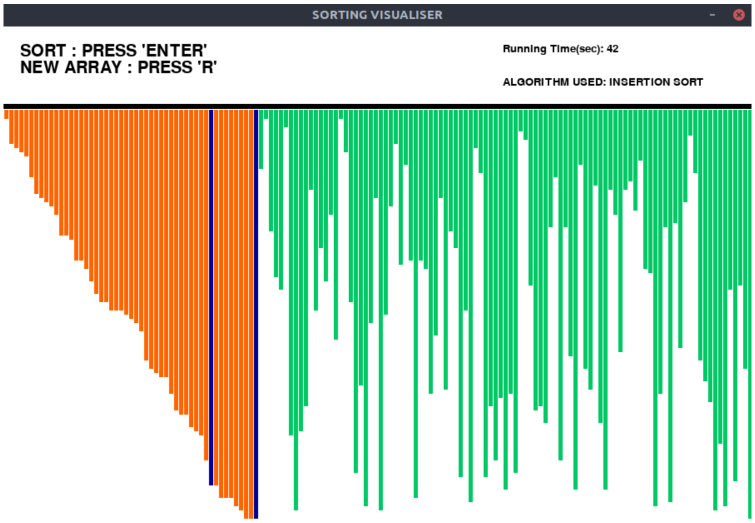


SORT : PRESS 'ENTER'
NEW ARRAY : PRESS 'R'

Running Time(sec): 16

ALGORITHM USED: INSERTION SORT

Sorting:

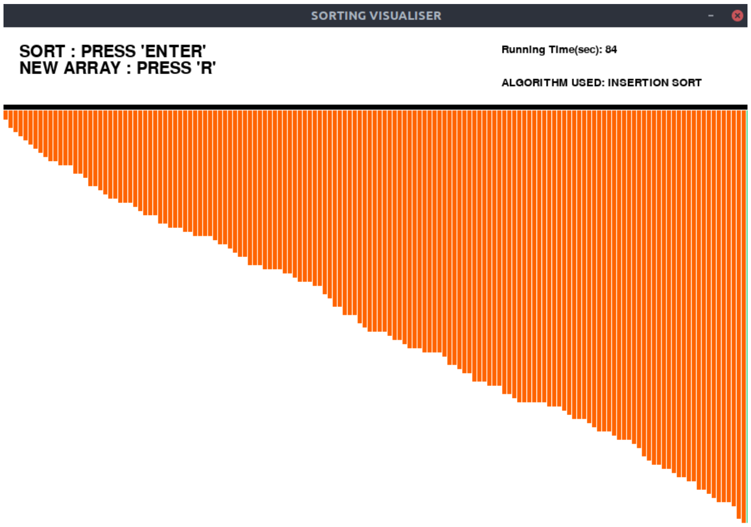


SORT : PRESS 'ENTER'
NEW ARRAY : PRESS 'R'

Running Time(sec): 42

ALGORITHM USED: INSERTION SORT

Final:



SORT : PRESS 'ENTER'
NEW ARRAY : PRESS 'R'

Running Time(sec): 84

ALGORITHM USED: INSERTION SORT

Please make sure to install the pygame library before running the below program.
Below is the implementation of the above visualizer:

Python3

```
# Python implementation of the
# Sorting visualiser: Insertion Sort

# Imports
import pygame
import random
import time
pygame.font.init()
startTime = time.time()
# Total window
screen = pygame.display.set_mode(
                        (900, 650)
                    )

# Title and Icon
pygame.display.set_caption(
            "SORTING VISUALISER"
            )

# Uncomment below lines for setting
# up the icon for the visuliser
# img = pygame.image.load('sorticon.png')
# pygame.display.set_icon(img)

# Boolean variable to run
# the program in while loop
run = True

# Window size and some initials
width = 900
length = 600
array =[0]*151
arr_clr =[(0, 204, 102)]*151
clr_ind = 0
clr =[(0, 204, 102), (255, 0, 0), \
       (0, 0, 153), (255, 102, 0)]
fnt = pygame.font.SysFont("comicsans", 30)
fnt1 = pygame.font.SysFont("comicsans", 20)

# Function to generate new Array
def generate_arr():
    for i in range(1, 151):
        arr_clr[i]= clr[0]
        array[i]= random.randrange(1, 100)

# Initially generate a array
generate_arr()

# Function to refill the
# updates on the window
def refill():
    screen.fill((255, 255, 255))
    draw()
    pygame.display.update()
    pygame.time.delay(10)

# Sorting Algorithm: Insertion sort
def insertionSort(array):

    for i in range(1, len(array)):
        pygame.event.pump()
        refill()
        key = array[i]
        arr_clr[i]= clr[2]
        j = i-1
        while j>= 0 and key<array[j]:
            arr_clr[j]= clr[2]
            array[j + 1]= array[j]
            refill()
            arr_clr[j]= clr[3]
            j = j-1
        array[j + 1]= key
        refill()
        arr_clr[i]= clr[0]

# Function to Draw the array values
def draw():
    # Text should be rendered
    txt = fnt.render("SORT: PRESS 'ENTER'", \
                                1, (0, 0, 0))
    # Position where text is placed
    screen.blit(txt, (20, 20))
    txt1 = fnt.render("NEW ARRAY: PRESS 'R'", \
                                1, (0, 0, 0))
    screen.blit(txt1, (20, 40))
    txt2 = fnt1.render("ALGORITHM USED:"\
           "INSERTION SORT", 1, (0, 0, 0))
    screen.blit(txt2, (600, 60))
    text3 = fnt1.render("Running Time(sec): "+\
            str(int(time.time() - startTime)), \
                                  1, (0, 0, 0))
    screen.blit(text3, (600, 20))
    element_width =(width-150)//150
    boundry_arr = 900 / 150
    boundry_grp = 550 / 100
    pygame.draw.line(screen, (0, 0, 0), (0, 95), \
                                  (900, 95), 6)

    # Drawing the array values as lines
    for i in range(1, 151):
        pygame.draw.line(screen, arr_clr[i], \
                   (boundry_arr * i-3, 100), \
                   (boundry_arr * i-3, \
     array[i]*boundry_grp + 100), element_width)

# Program should be run
# continuously to keep the window open
while run:
    # background
    screen.fill((255, 255, 255))

    # Event handler stores all event
    for event in pygame.event.get():

        # If we click Close button in window
        if event.type == pygame.QUIT:
            run = False
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_r:
                generate_arr()
            if event.key == pygame.K_RETURN:
                insertionSort(array)
    draw()
    pygame.display.update()

pygame.quit()
```

Output:

00:00 01:09

M mano... 3

Previous Article Next Article

Sorting algorithm visualization : Heap Sort

Binary Search Visualization using Pygame in Python

Similar Reads

Sorting Algorithm Visualization : Merge Sort

The human brain can easily process visuals instead of long codes to understand the algorithms. In this article, a program that program visualizes the Merge sort...

3 min read

Sorting algorithm visualization : Heap Sort

An algorithm like Heap sort can be understood easily by visualizing. In this article, a program that visualizes the Heap Sort Algorithm has been implemented. The...

4 min read

Sorting Algorithm Visualization : Quick Sort

An algorithm like Quicksort algorithm is hard to understand theoretically. We can understand easily by visualizing such kind of algorithms. In this article, a progra...

3 min read

Sorting by combining Insertion Sort and Merge Sort algorithms

Insertion sort: The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in th...

2 min read

What is the stupidest sorting algorithm? (Worst Sorting Algorithm)

Bogo sort stands out as the undisputed champion of stupidity. Unlike other sorting algorithms that follow a structured approach, Bogo sort relies on sheer...

2 min read

Insertion Sort Visualization using Matplotlib in Python

Prerequisites: Insertion Sort, Using Matplotlib for Animations Visualizing algorithms makes it easier to understand them by analyzing and comparing the...

3 min read

Insertion Sort Visualization using JavaScript

Insertion sort is a simple sorting algorithm in which values from the unsorted part are picked and placed at the correct position in the sorted part. In order to know...

5 min read

Comparison among Bubble Sort, Selection Sort and Insertion Sort

Bubble Sort, Selection Sort, and Insertion Sort are simple sorting algorithms that are commonly used to sort small datasets or as building blocks for more comple...

15 min read

Sorting Algorithms Visualization | Selection Sort

The human brain can easily process visuals in spite of long codes to understand the algorithms. In this article, Selection Sort visualization has been implemented...

6 min read

Sorting an Array in Bash using Insertion Sort

Given an array, arr[] of size N, the task is to sort the array in ascending order using Insertion Sort in bash scripting. Examples: Input: arr[] = {9, 7, 2, 5}Output: 2 5 7...

2 min read

Article Tags : Algorithms DSA Project Python +4 More

Practice Tags : Algorithms python Sorting

Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305

Company

About Us

Legal

In Media

Contact Us

Advertise with us

GFG Corporate Solution

Placement Training Program

GeeksforGeeks Community

DSA

Data Structures

Algorithms

DSA for Beginners

Basic DSA Problems

DSA Roadmap

Top 100 DSA Interview Problems

DSA Roadmap by Sandeep Jain

All Cheat Sheets

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

Languages

Python

Java

C++

PHP

GoLang

SQL

R Language

Android Tutorial

Tutorials Archive

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

NLP

Deep Learning

Python Tutorial

Python Programming Examples

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Django

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects