

Itisha Desai

Sem – 5

Branch – Cyber Security

Batch – CSE54

Enrollment No. – 22162171006

## **Subject : Algorithm Analysis and Design**

### **Practical 3**

NextMid Technology is an American food company that manufactures, markets, and distributes spices, seasoning mixes, condiments, and other flavoring products for the industrial, restaurant, institutional, and home markets, they are having some number quantity of different categories item food, kindly help them to sort data using any three sorting methods and determine the time required to sort the elements. Repeat the experiment for different values of  $n$ , the number of elements in the list to be sorted and plot a graph of the comparison between them.

Design the algorithm for the same and implement using the programming language of your choice. Make comparative analysis for various use cases & input size.

## Questions:

What is the best, average and worst case analysis of algorithms?

Which are different asymptotic notations? What is their use?

What is the time complexity of above 3 sorting algorithms in all cases?

## Code :

### Python File- main.py

```
from flask import Flask, render_template
```

```
import time
```

```
import random
```

```
import plotly.graph_objs as go
```

```
import plotly.offline as pyo
```

```
app = Flask(__name__)
```

```
# Sorting algorithm implementations
```

```
def selection_sort(arr):
```

```
    n = len(arr)
```

```
    for i in range(n):
```

```
        min_idx = i
```

```
        for j in range(i + 1, n):
```

```
            if arr[j] < arr[min_idx]:
```

```
                min_idx = j
```

```
    arr[i], arr[min_idx] = arr[min_idx], arr[i]
```

```

def insertion_sort(arr):
    n = len(arr)
    for i in range(1, n):
        key = arr[i]
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
# Function to measure execution time
def measure_time(sort_function, arr):
    start_time = time.time()
    sort_function(arr.copy()) # Use a copy to prevent modifying the original
                                array
    end_time = time.time()
    return end_time - start_time
# Flask route for the homepage
@app.route('/')
def index():
    sizes = [100, 300, 600, 1000, 8000] # Different input sizes
    selection_times = []

```

```

insertion_times = []
bubble_times = []

# Measure sorting times for different array sizes
for size in sizes:
    arr = random.sample(range(size * 10), size) # Generate random list
    selection_times.append(round(measure_time(selection_sort, arr), 5))
    insertion_times.append(round(measure_time(insertion_sort, arr), 5))
    bubble_times.append(round(measure_time(bubble_sort, arr), 5))

# Plotly traces for visualization
trace1 = go.Scatter(x=sizes, y=selection_times, mode='lines+markers',
name='Selection Sort')
trace2 = go.Scatter(x=sizes, y=insertion_times, mode='lines+markers',
name='Insertion Sort')
trace3 = go.Scatter(x=sizes, y=bubble_times, mode='lines+markers',
name='Bubble Sort')

layout = go.Layout(title='Comparison of Sorting Algorithms',
                    xaxis=dict(title='Number of Elements'),
                    yaxis=dict(title='Time Taken (seconds)'))

# Generate the plot
fig = go.Figure(data=[trace1, trace2, trace3], layout=layout)
plot_html = pyo.plot(fig, output_type='div', include_plotlyjs=True)

# Render the HTML template with plot and data
return render_template('index.html', plot=plot_html, sizes=sizes,
                      selection_times=selection_times,
                      insertion_times=insertion_times,
                      bubble_times=bubble_times)

```

```
        bubble_times=bubble_times)

# Run the app

if __name__ == '__main__':
    app.run(debug=True)
```

## HTML File - index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Sorting Algorithm Comparison</title>

</head>

<body>

    <h2>Comparison of Sorting Algorithms</h2>

    <div>{{ plot|safe }}</div> <!-- Render the Plotly plot -->

    <h3>Sorting Times for Different Input Sizes</h3>

    <table border="1">

        <thead>

            <tr>

                <th>Number of Elements</th>

                <th>Selection Sort Time (s)</th>

                <th>Insertion Sort Time (s)</th>

                <th>Bubble Sort Time (s)</th>

            </tr>
```

```

</thead>
<tbody>
    {% for i in range(sizes|length) %}
    <tr>
        <td>{{ sizes[i] }}</td>
        <td>{{ selection_times[i] }}</td>
        <td>{{ insertion_times[i] }}</td>
        <td>{{ bubble_times[i] }}</td>
    </tr>
    {% endfor %}
</tbody>
</table>
</body>
</html>

```

## Output :

