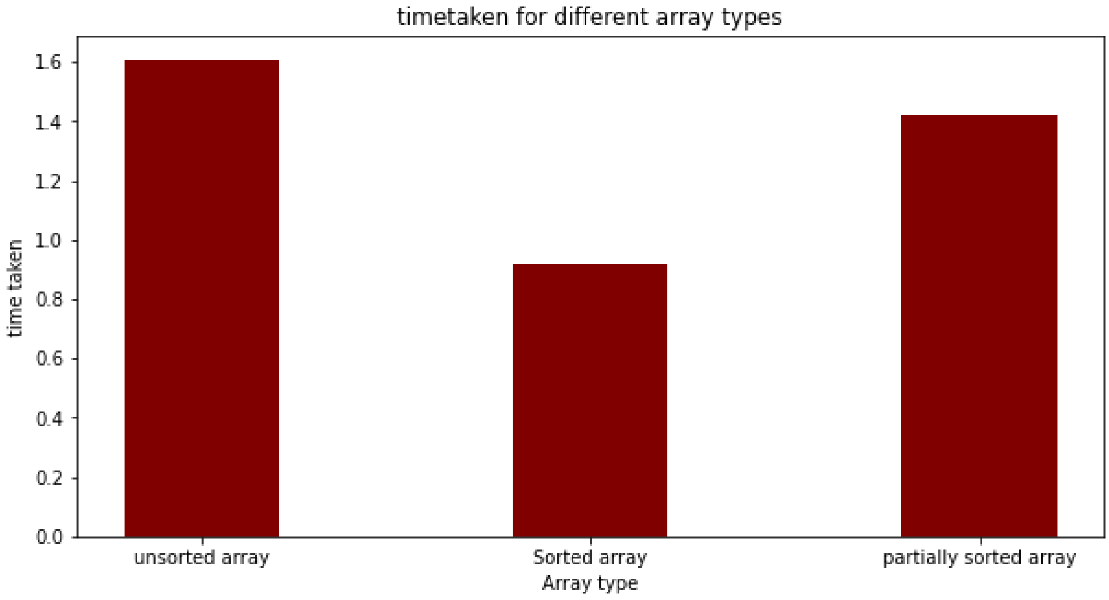
```
import time
import random as rn
import matplotlib.pyplot as plt
```

Bubble Sort

```
def bubbleSort(arr):
    for i in range(len(arr)):
        for j in range(0, len(arr)-i-1):
            if arr[j] > arr[j+1] :
                temp=arr[j]
                arr[j]=arr[j+1]
                arr[j+1]=temp



arr=[]
runTime1=[]
n=5000

for i in range (n):
    arr.append(rn.randint(1,1000000))

#sorting random array
arr1=[]
for i in range (n):
    arr1.append(arr[i])

start=time.time()
bubbleSort(arr1)
stop=time.time()

runTime1.append(stop-start)

#sorting sorted array

start=time.time()
bubbleSort(arr1)
stop=time.time()

runTime1.append(stop-start)

#sorting partially sorted array

arr2=[]

for i in range(n//2):
    arr2.append(arr[i])
```

```
bubbleSort(arr2)

arr3=[]

for i in range (n):
    arr3.append(arr[i])

for i in range (n//2):
    arr3[i]=arr2[i]

start=time.time()
bubbleSort(arr3)
stop=time.time()

runTime1.append(stop-start)


import numpy as np

data = {'unsorted array':runTime1[0], 'Sorted array':runTime1[1],
'partially sorted array':runTime1[2]}
arrayType = list(data.keys())
time = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(arrayType, time, color ='maroon',width = 0.4)

plt.xlabel("Array type")
plt.ylabel("time taken")
plt.title("timetaken for different array types")
plt.show()
```

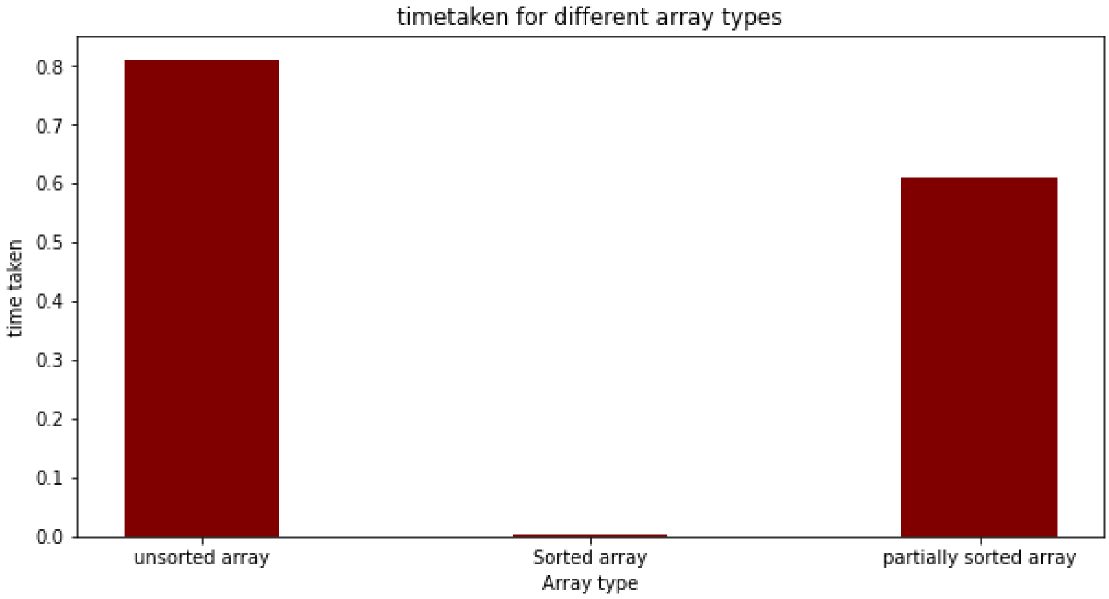


Insertion Sort

```
def insertionSort(arr):
    for i in range(1,len(arr)):
        key=arr[i]
        j=i-1
        while key<arr[j] and j>=0:
            arr[j+1]=arr[j]
            j-=1
        arr[j+1]=key
    return
```

```
import time

arr=[]
runTime2=[]
n=5000

for i in range (n):
    arr.append(rn.randint(1,1000000))

#sorting random array
arr1=[]
for i in range (n):
    arr1.append(arr[i])

start=time.time()
insertionSort(arr1)
stop=time.time()

runTime2.append(stop-start)
```

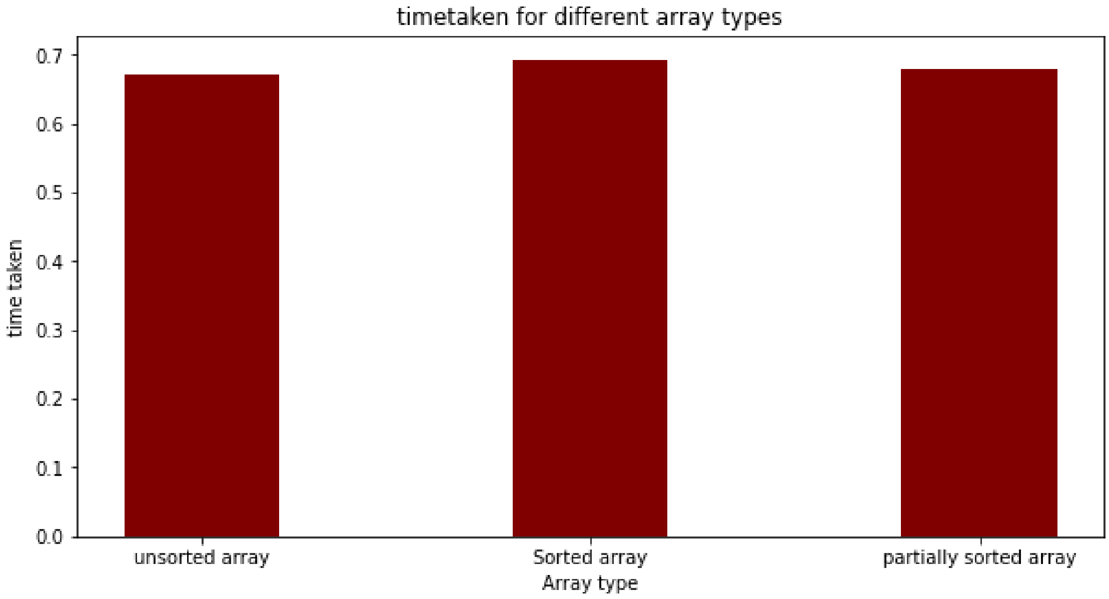
```
#sorting sorted array

start=time.time()
insertionSort(arr1)
stop=time.time()

runTime2.append(stop-start)

#sorting partially sorted array

arr2=[]

for i in range(n//2):
    arr2.append(arr[i])

insertionSort(arr2)

arr3=[]

for i in range (n):
    arr3.append(arr[i])

for i in range (n//2):
    arr3[i]=arr2[i]

start=time.time()
insertionSort(arr3)
stop=time.time()

runTime2.append(stop-start)


data = {'unsorted array':runTime2[0], 'Sorted array':runTime2[1],
'partially sorted array':runTime2[2]}
arrayType = list(data.keys())
time = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(arrayType, time, color ='maroon',width = 0.4)

plt.xlabel("Array type")
plt.ylabel("time taken")
plt.title("timetaken for different array types")
plt.show()
```



Selection Sort

```
def selectionSort(arr):
    for i in range(0,len(arr)-1):
        min_index=i
        for j in range(i,len(arr)):
            if arr[min_index]>arr[j]:
                min_index=j
        temp=arr[min_index]
        arr[min_index]=arr[i]
        arr[i]=temp
    return

import time

arr=[]
runTime3=[]
n=5000

for i in range (n):
    arr.append(rn.randint(1,1000000))

#sorting random array
arr1=[]
for i in range (n):
    arr1.append(arr[i])

start=time.time()
selectionSort(arr1)
stop=time.time()
```

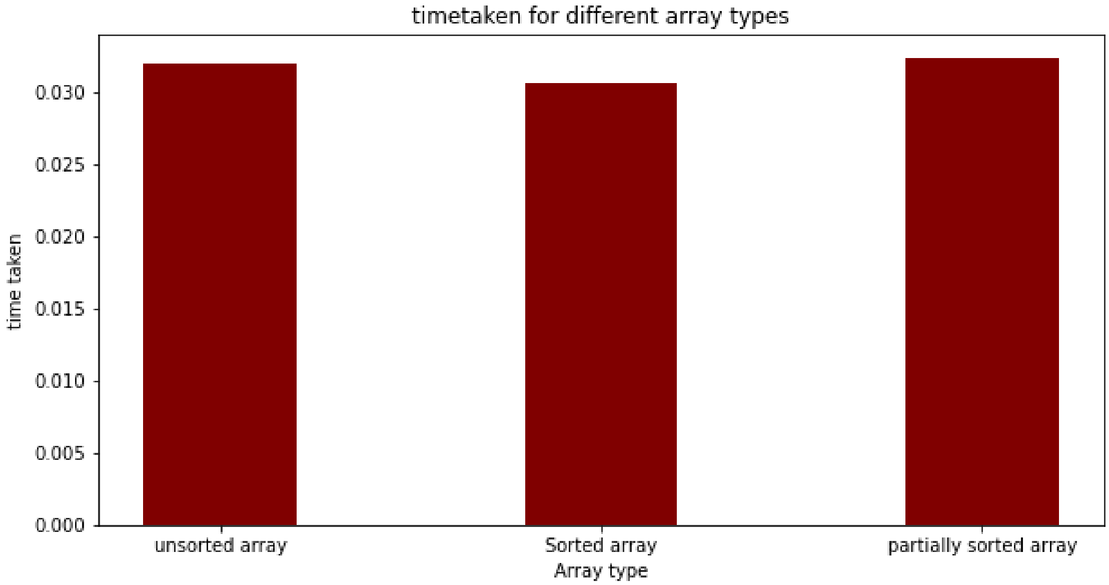
```python
runTime3.append(stop-start)

#sorting sorted array

start=time.time()
selectionSort(arr1)
stop=time.time()

runTime3.append(stop-start)

#sorting partially sorted array

arr2=[]

for i in range(n//2):
    arr2.append(arr[i])

selectionSort(arr2)

arr3=[]

for i in range (n):
    arr3.append(arr[i])

for i in range (n//2):
    arr3[i]=arr2[i]

start=time.time()
selectionSort(arr3)
stop=time.time()

runTime3.append(stop-start)


data = {'unsorted array':runTime3[0], 'Sorted array':runTime3[1],
'partially sorted array':runTime3[2]}
arrayType = list(data.keys())
time = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(arrayType, time, color ='maroon',width = 0.4)

plt.xlabel("Array type")
plt.ylabel("time taken")
plt.title("timetaken for different array types")
plt.show()
```



```
#print(runTime)
```

Merge Sort

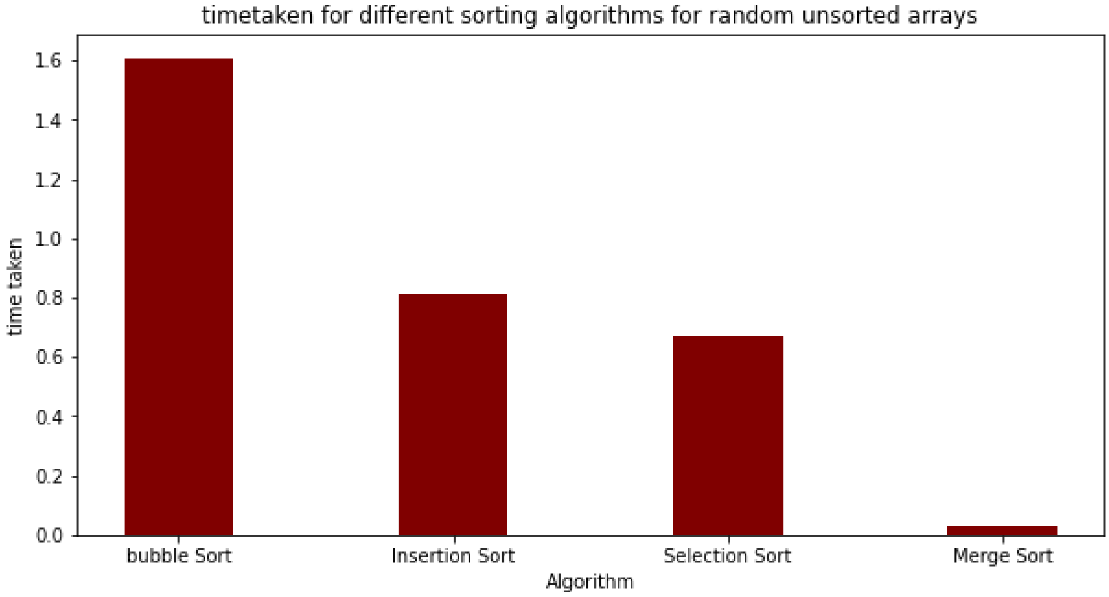
```
def merge(a,b,arr):
    i=0
    j=0
    k=0

    while (i<len(a) and j<len(b)):
        if (a[i]<=b[j]):
            arr[k]=a[i]
            i+=1
        else:
            arr[k]=b[j]
            j+=1
        k+=1

    while (i<len(a)):
        arr[k]=a[i]
        i+=1
        k+=1

    while (j<len(b)):
        arr[k]=b[j]
        j+=1
        k+=1

def mergeSort(arr):
    if len(arr)<=1:
        return
    mid=len(arr)//2
    low=arr[:mid]
    high=arr[mid:]

    mergeSort(low)
    mergeSort(high)
    merge(low,high,arr)

import time

arr=[]
runTime4=[]
n=5000

for i in range (n):
    arr.append(rn.randint(1,1000000))

#sorting random array
arr1=[]
for i in range (n):
    arr1.append(arr[i])

start=time.time()
mergeSort(arr1)
stop=time.time()

runTime4.append(stop-start)

#sorting sorted array

start=time.time()
mergeSort(arr1)
stop=time.time()

runTime4.append(stop-start)

#sorting partially sorted array

arr2=[]

for i in range(n//2):
    arr2.append(arr[i])

mergeSort(arr2)

arr3=[]

for i in range (n):
    arr3.append(arr[i])

for i in range (n//2):
    arr3[i]=arr2[i]

start=time.time()
mergeSort(arr3)
stop=time.time()

runTime4.append(stop-start)


data = {'unsorted array':runTime4[0], 'Sorted array':runTime4[1],
'partially sorted array':runTime4[2]}
arrayType = list(data.keys())
time = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(arrayType, time, color ='maroon',width = 0.4)

plt.xlabel("Array type")
plt.ylabel("time taken")
plt.title("timetaken for different array types")
plt.show()
```



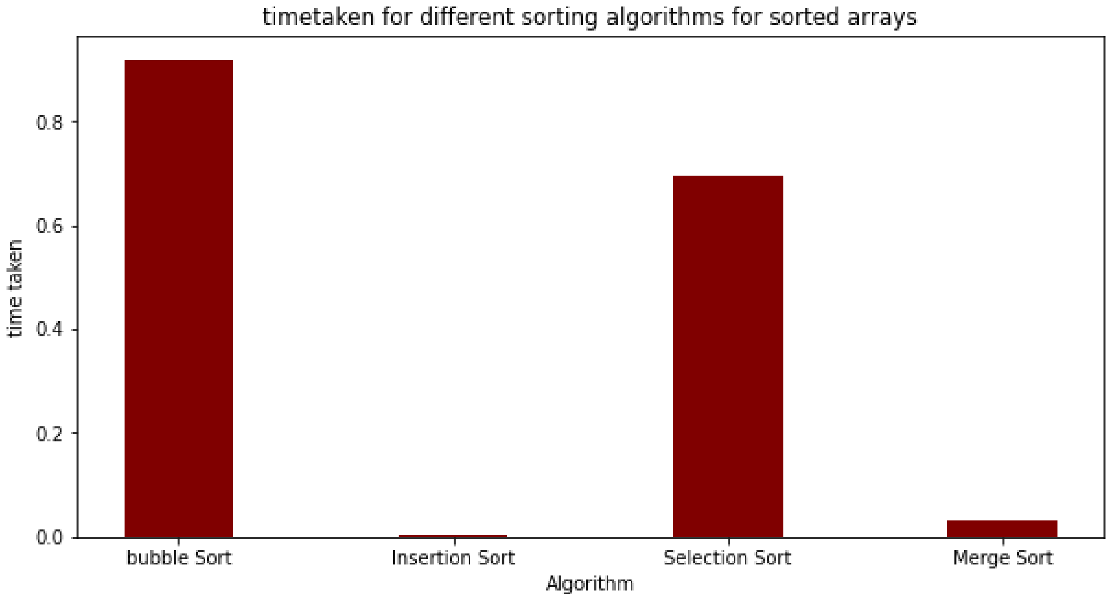
Time taken for different algorithms for same array type

```
data = {'bubble Sort':runTime1[0], 'Insertion Sort':runTime2[0],
'Selection Sort':runTime3[0], 'Merge Sort':runTime4[0]}
```

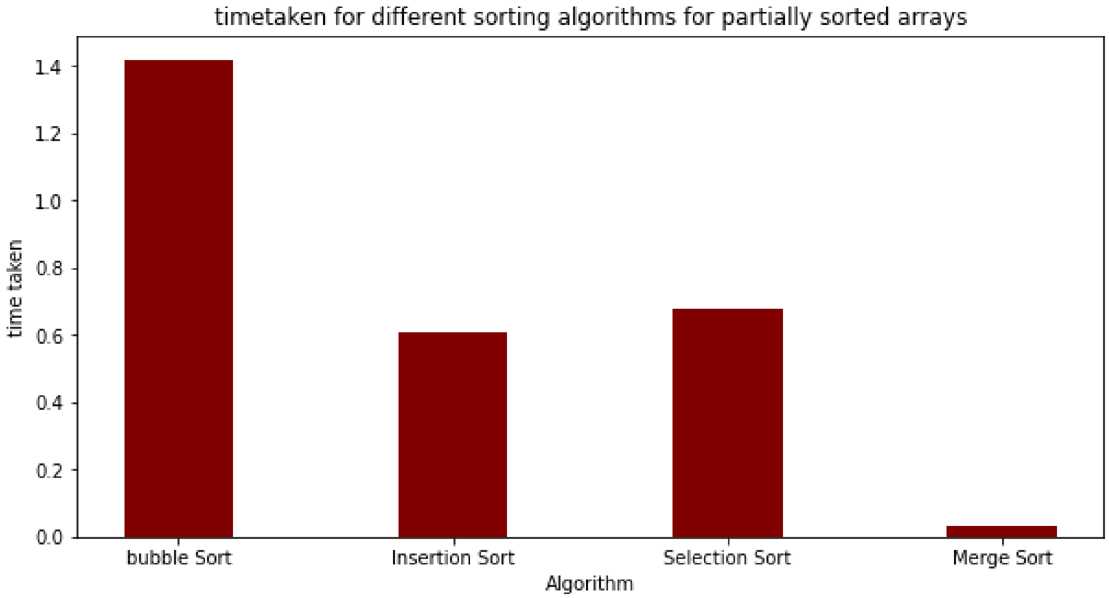
```python
arrayType = list(data.keys())
time = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(arrayType, time, color ='maroon',width = 0.4)

plt.xlabel("Algorithm")
plt.ylabel("time taken")
plt.title("timetaken for different sorting algorithms for random unsorted arrays")
plt.show()
```



```python
data = {'bubble Sort':runTime1[1], 'Insertion Sort':runTime2[1], 'Selection Sort':runTime3[1], 'Merge Sort':runTime4[1]}
arrayType = list(data.keys())
time = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(arrayType, time, color ='maroon',width = 0.4)

plt.xlabel("Algorithm")
plt.ylabel("time taken")
plt.title("timetaken for different sorting algorithms for sorted arrays")
plt.show()
```



```
data = {'bubble Sort':runTime1[2], 'Insertion Sort':runTime2[2],
'Selection Sort':runTime3[2], 'Merge Sort':runTime4[2]}
arrayType = list(data.keys())
time = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(arrayType, time, color ='maroon',width = 0.4)

plt.xlabel("Algorithm")
plt.ylabel("time taken")
plt.title("timetaken for different sorting algorithms for partially
sorted arrays")
plt.show()
```


timetaken for different sorting algorithms for partially sorted arrays
time taken
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
bubble Sort
Insertion Sort
Selection Sort
Merge Sort
Algorithm