```python
def partition(low,high,arr):

    p=low-1
    pivot=arr[high]

    if low<high:
        for i in range(low,high+1):
            if arr[i]<=pivot:
                temp=arr[i]
                arr[i]=arr[p+1]
                arr[p+1]=temp
                p+=1

    return p

def quickSort(start,end,arr):
    if start<end:
        i=partition(start,end,arr)
        quickSort(start,i-1,arr)
        quickSort(i+1,end,arr)



arr=[9,8,7,6,5,4,3,2,1,0]
quickSort(0,len(arr)-1,arr)
print(arr)

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

import time
import random as rn
import matplotlib.pyplot as plt
import numpy as np

n=[1000,2000,3000,4000,5000,6000,7000,8000,9000,10000]
runTime=[]
arr=[]

for i in range(1000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTime.append(stop-start)



arr=[]
```

```python
for i in range(2000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTime.append(stop-start)

arr=[]

for i in range(3000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTime.append(stop-start)

arr=[]

for i in range(4000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTime.append(stop-start)

arr=[]

for i in range(5000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTime.append(stop-start)

arr=[]

for i in range(6000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTime.append(stop-start)

arr=[]

for i in range(7000):
```

```
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTime.append(stop-start)

arr=[]

for i in range(8000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTime.append(stop-start)

arr=[]

for i in range(9000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTime.append(stop-start)

arr=[]

for i in range(10000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTime.append(stop-start)

#print(runTime)
```

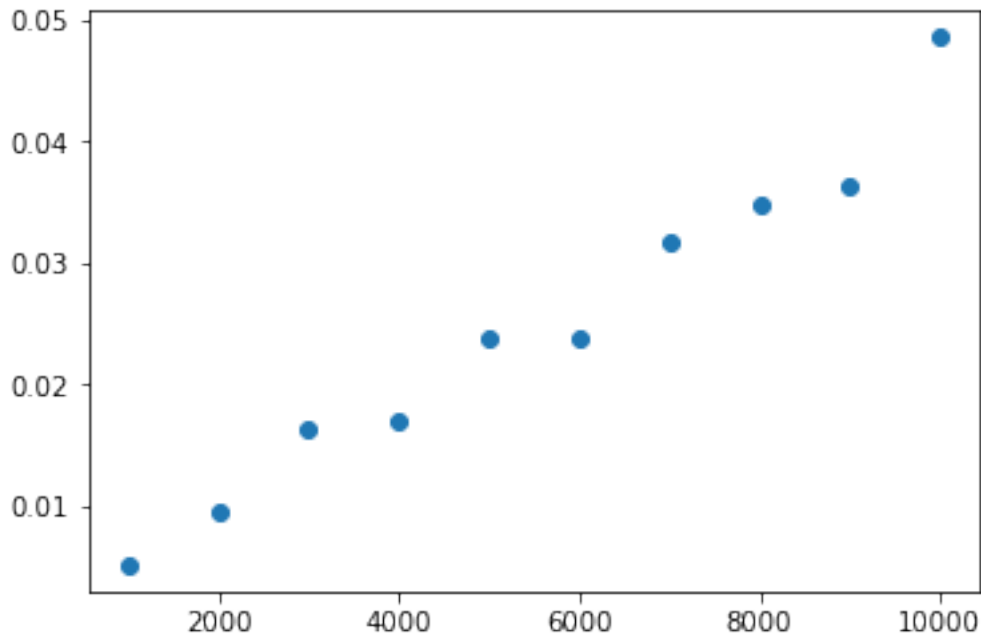## Scatterplot for runtime vs size of array

```
plt.scatter(n,runTime)
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

Variance of runtime for a fixed array size

```python
import statistics


runTimeVariance=[]

arr=[]
for i in range(5000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTimeVariance.append(stop-start)

arr=[]
for i in range(5000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTimeVariance.append(stop-start)

arr=[]
for i in range(5000):
    arr.append(rn.randint(0,1000000))
```

```python
start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTimeVariance.append(stop-start)

arr=[]
for i in range(5000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTimeVariance.append(stop-start)

arr=[]
for i in range(5000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTimeVariance.append(stop-start)

arr=[]
for i in range(5000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTimeVariance.append(stop-start)

arr=[]
for i in range(5000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTimeVariance.append(stop-start)

arr=[]
for i in range(5000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTimeVariance.append(stop-start)
```

```python
arr=[]
for i in range(5000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTimeVariance.append(stop-start)

arr=[]
for i in range(5000):
    arr.append(rn.randint(0,1000000))

start=time.time()
quickSort(0,len(arr)-1,arr)
stop=time.time()
runTimeVariance.append(stop-start)

variance=statistics.variance(runTimeVariance)

print("The variance of runtime for an array size of 5000 is
",variance)
```

```
The variance of runtime for an array size of 5000 is
4.046987210194351e-05
```

The runtime depends on the position of the pivot and since the pivot changes as the numbers in the array are changed a variance in runtime is observed for the same size of array