Splitting into 2 arrays

```
def merge2arrays(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 mergeSort2arrays(arr):
    #print(arr)
    if len(arr)<=1:
        return

    mid=len(arr)//2
    low=arr[:mid]
    high=arr[mid:]

    mergeSort2arrays(low)
    mergeSort2arrays(high)
    merge2arrays(low,high,arr)
```

Splitting into 3 arrays

```
def merge3arrays(a,b,c,arr):

    x=0
    y=0
    z=0
    k=0
```

```
while(x<len(a) and y<len(b) and z<len(c)):

    if(a[x]<=b[y] and a[x]<=c[z]):
        arr[k]=a[x]
        x+=1

    elif(b[y]<=a[x] and b[y]<=c[z]):
        arr[k]=b[y]
        y+=1

    else:
        arr[k]=c[z]
        z+=1

    k+=1

while (x<len(a) and y<len(b) and z>=len(c)):

    if(a[x]<=b[y]):
        arr[k]=a[x]
        x+=1

    else:
        arr[k]=b[y]
        y+=1

    k+=1

while (x>=len(a) and y<len(b) and z<len(c)):
    if(b[y]<=c[z]):
        arr[k]=b[y]
        y+=1

    else:
        arr[k]=c[z]
        z+=1

    k+=1

while (x<len(a) and y>=len(b) and z<len(c)):
    if(a[x]<=c[z]):
        arr[k]=a[x]
        x+=1

    else:
        arr[k]=c[z]
        z+=1

    k+=1
```

```
    while (x<len(a) and y>=len(b) and z>=len(c)):
        arr[k]=a[x]
        x+=1
        k+=1

    while (x>=len(a) and y<len(b) and z>=len(c)):
        arr[k]=b[y]
        y+=1
        k+=1

    while (x>=len(a) and y>=len(b) and z<len(c)):
        arr[k]=c[z]
        z+=1
        k+=1



def swap(arr):
    temp=arr[0]
    arr[0]=arr[1]
    arr[1]=temp

def  mergeSort3arrays(arr):
   # print(arr)
    if (len(arr)<=1):
        return
    elif(len(arr)==2):
        if(arr[0]>arr[1]):
            swap(arr)
        return


    t=len(arr)//3

    arr1=arr[:t]
    arr2=arr[t:2*t]
    arr3=arr[2*t:]

    mergeSort3arrays(arr1)
    mergeSort3arrays(arr2)
    mergeSort3arrays(arr3)
    merge3arrays(arr1,arr2,arr3,arr)


arr=[2,0,88,44,5]

mergeSort2arrays(arr)
print(arr)
```

```
[0, 2, 5, 44, 88]
```

```
arr=[2,0,88,44,5]
#merge3arrays(arr1,arr2,arr3,arr)
mergeSort3arrays(arr)
print(arr)
```

```
[0, 2, 5, 44, 88]
```

Splitting into 4 arrays

```
def merge4arrays(a,b,c,d,arr):

    w=0
    x=0
    y=0
    z=0
    k=0

    while (w<len(a) and x<len(b) and y<len(c) and z <len(d)):
        if(a[w]<=b[x] and a[w]<=c[y] and a[w] <= d[z]):
            arr[k]=a[w]
            w+=1

        elif(b[x]<=a[w] and b[x]<=c[y] and b[x] <= d[z]):
            arr[k]=b[x]
            x+=1

        elif(c[y]<=b[x] and c[y]<=a[w] and c[y] <= d[z]):
            arr[k]=c[y]
            y+=1

        elif(d[z]<=b[x] and d[z]<=c[y] and d[z] <= a[w]):
            arr[k]=d[w]
            z+=1

        k+=1

    while (w <len(a) and x <len(b) and y <len(c) and z>= len(d)):
        if(a[w]<=b[x] and a[w]<=c[y]):
            arr[k]=a[w]
            w+=1

        elif(b[x]<=a[w] and b[x]<=c[y]):
            arr[k]=b[x]
            x+=1

        else:
            arr[k]=c[y]
            y+=1

        k+=1
```

```
while (w <len(a) and x <len(b) and y >=len(c) and z< len(d)):
    if(a[w]<=b[x] and a[w]<=d[z]):
        arr[k]=a[w]
        w+=1

    elif(b[x]<=a[w] and b[x]<=d[z]):
        arr[k]=b[x]
        x+=1

    else:
        arr[k]=d[z]
        z+=1

    k+=1

while (w <len(a) and x >=len(b) and y <len(c) and z< len(d)):
    if(a[w]<=c[y] and a[w]<=d[z]):
        arr[k]=a[w]
        w+=1

    elif(c[y]<=a[w] and c[y]<=d[z]):
        arr[k]=c[y]
        y+=1

    else:
        arr[k]=d[z]
        z+=1

    k+=1

while (w >=len(a) and x <len(b) and y <len(c) and z< len(d)):
    if(b[x]<=c[y] and b[x]<=d[z]):
        arr[k]=b[x]
        x+=1

    elif(c[y]<=b[x] and c[y]<=d[z]):
        arr[k]=c[y]
        y+=1

    else:
        arr[k]=d[z]
        z+=1

    k+=1

while (w <len(a) and x <len(b) and y >=len(c) and z>= len(d)):
    if(a[w]<=b[x]):
        arr[k]=a[w]
        w+=1
```

```
            k+=1
        else:
            arr[k]=b[x]
            x+=1
            k+=1

    while (w <len(a) and x >=len(b) and y <len(c) and z>= len(d)):
        if(a[w]<=c[y]):
            arr[k]=a[w]
            w+=1
            k+=1
        else:
            arr[k]=c[y]
            y+=1
            k+=1

    while (w <len(a) and x >=len(b) and y >=len(c) and z< len(d)):
        if(a[w]<=d[z]):
            arr[k]=a[w]
            w+=1
            k+=1
        else:
            arr[k]=d[z]
            z+=1
            k+=1

    while (w >=len(a) and x <len(b) and y <len(c) and z>= len(d)):
        if(b[x]<=c[y]):
            arr[k]=b[x]
            x+=1
            k+=1
        else:
            arr[k]=c[y]
            y+=1
            k+=1

    while (w >=len(a) and x <len(b) and y >=len(c) and z< len(d)):
        if(b[x]<=d[z]):
            arr[k]=b[x]
            x+=1
            k+=1
        else:
            arr[k]=d[z]
            z+=1
            k+=1

    while (w >=len(a) and x >=len(b) and y <len(c) and z< len(d)):
        if(c[y]<=d[z]):
            arr[k]=c[y]
            y+=1
```

```
            k+=1
        else:
            arr[k]=d[z]
            z+=1
            k+=1

    while (w <len(a) and x >=len(b) and y >=len(c) and z>= len(d)):
        arr[k]=a[w]
        w+=1
        k+=1

    while (w >=len(a) and x <len(b) and y >=len(c) and z>= len(d)):
        arr[k]=b[x]
        x+=1
        k+=1

    while (w >=len(a) and x >=len(b) and y <len(c) and z>= len(d)):
        arr[k]=c[y]
        y+=1
        k+=1

    while (w >=len(a) and x >=len(b) and y >=len(c) and z< len(d)):
        arr[k]=d[z]
        z+=1
        k+=1

def mergeSort4arrays(arr):
    #print(arr)
    if (len(arr)<=1):
        return
    elif(len(arr)==2):
        if(arr[0]>arr[1]):
            swap(arr)
        return
    elif(len(arr)==3):
        mergeSort3arrays(arr)
        return


    t=len(arr)//4

    arr1=arr[:t]
    arr2=arr[t:2*t]
    arr3=arr[2*t:3*t]
    arr4=arr[3*t:]

    mergeSort4arrays(arr1)
    mergeSort4arrays(arr2)
    mergeSort4arrays(arr3)
```

```
    mergeSort4arrays(arr4)
    merge4arrays(arr1,arr2,arr3,arr4,arr)

arr=[2,0,88,44,5]

mergeSort4arrays(arr)
print(arr)
```

```
[0, 2, 5, 44, 88]
```

```
import random as rn
import time

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

t2=[]
t3=[]
t4=[]
```

For array size 1000

```
arr=[]
arr1=[]
arr2=[]
arr3=[]
for i in range(1000):
    a=rn.randint(0,100000)
    arr.append(a)

for i in range(1000):
    arr1.append(arr[i])
    arr2.append(arr[i])
    arr3.append(arr[i])

start=time.time()
mergeSort2arrays(arr1)
stop=time.time()

t2.append(stop - start)


start=time.time()
mergeSort3arrays(arr2)
stop=time.time()

t3.append(stop - start)


start=time.time()
mergeSort4arrays(arr3)
stop=time.time()
```

```
t4.append(stop - start)

print(t2,t2,t4)
```

```
[0.007973194122314453] [0.007973194122314453] [0.0069010257720947266]
```

For array size 2000

```
#print(arr,arr1,arr2,arr3)

arr=[]
arr1=[]
arr2=[]
arr3=[]
for i in range(2000):
    a=rn.randint(1,100000)
    arr.append(a)

for i in range(2000):
    arr1.append(arr[i])
    arr2.append(arr[i])
    arr3.append(arr[i])

start=time.time()
mergeSort2arrays(arr1)
stop=time.time()

t2.append(stop - start)


start=time.time()
mergeSort3arrays(arr2)
stop=time.time()

t3.append(stop - start)


start=time.time()
mergeSort4arrays(arr3)
stop=time.time()

t4.append(stop - start)
```

For array size 3000

```
arr=[]
arr1=[]
arr2=[]
arr3=[]
for i in range(3000):
    a=rn.randint(0,100000)
    arr.append(a)

for i in range(3000):
    arr1.append(arr[i])
    arr2.append(arr[i])
    arr3.append(arr[i])

start=time.time()
mergeSort2arrays(arr1)
stop=time.time()

t2.append(stop - start)


start=time.time()
mergeSort3arrays(arr2)
stop=time.time()

t3.append(stop - start)


start=time.time()
mergeSort4arrays(arr3)
stop=time.time()

t4.append(stop - start)
```

For array size 4000

```
arr=[]
arr1=[]
arr2=[]
arr3=[]
for i in range(4000):
    a=rn.randint(0,100000)
    arr.append(a)

for i in range(4000):
    arr1.append(arr[i])
    arr2.append(arr[i])
    arr3.append(arr[i])

start=time.time()
mergeSort2arrays(arr1)
```

```
stop=time.time()

t2.append(stop - start)


start=time.time()
mergeSort3arrays(arr2)
stop=time.time()

t3.append(stop - start)


start=time.time()
mergeSort4arrays(arr3)
stop=time.time()

t4.append(stop - start)
```

For array size 5000

```
arr=[]
arr1=[]
arr2=[]
arr3=[]
for i in range(5000):
    a=rn.randint(0,100000)
    arr.append(a)

for i in range(5000):
    arr1.append(arr[i])
    arr2.append(arr[i])
    arr3.append(arr[i])

start=time.time()
mergeSort2arrays(arr1)
stop=time.time()

t2.append(stop - start)


start=time.time()
mergeSort3arrays(arr2)
stop=time.time()

t3.append(stop - start)


start=time.time()
mergeSort4arrays(arr3)
stop=time.time()
```

```
t4.append(stop - start)
```

For array size 6000

```
arr=[]
arr1=[]
arr2=[]
arr3=[]
for i in range(6000):
    a=rn.randint(0,100000)
    arr.append(a)

for i in range(6000):
    arr1.append(arr[i])
    arr2.append(arr[i])
    arr3.append(arr[i])

start=time.time()
mergeSort2arrays(arr1)
stop=time.time()

t2.append(stop - start)


start=time.time()
mergeSort3arrays(arr2)
stop=time.time()

t3.append(stop - start)


start=time.time()
mergeSort4arrays(arr3)
stop=time.time()

t4.append(stop - start)
```

For array size 7000

```
arr=[]
arr1=[]
arr2=[]
arr3=[]
for i in range(7000):
    a=rn.randint(0,100000)
    arr.append(a)

for i in range(7000):
    arr1.append(arr[i])
```

```
    arr2.append(arr[i])
    arr3.append(arr[i])

start=time.time()
mergeSort2arrays(arr1)
stop=time.time()

t2.append(stop - start)


start=time.time()
mergeSort3arrays(arr2)
stop=time.time()

t3.append(stop - start)


start=time.time()
mergeSort4arrays(arr3)
stop=time.time()

t4.append(stop - start)

#print(t2,t3,t4)
```

For array size 8000

```
arr=[]
arr1=[]
arr2=[]
arr3=[]
for i in range(8000):
    a=rn.randint(0,100000)
    arr.append(a)

for i in range(8000):
    arr1.append(arr[i])
    arr2.append(arr[i])
    arr3.append(arr[i])

start=time.time()
mergeSort2arrays(arr1)
stop=time.time()

t2.append(stop - start)


start=time.time()
mergeSort3arrays(arr2)
stop=time.time()
```

```
t3.append(stop - start)


start=time.time()
mergeSort4arrays(arr3)
stop=time.time()

t4.append(stop - start)
```

For array size 9000

```
arr=[]
arr1=[]
arr2=[]
arr3=[]
for i in range(9000):
    a=rn.randint(0,100000)
    arr.append(a)

for i in range(9000):
    arr1.append(arr[i])
    arr2.append(arr[i])
    arr3.append(arr[i])

start=time.time()
mergeSort2arrays(arr1)
stop=time.time()

t2.append(stop - start)


start=time.time()
mergeSort3arrays(arr2)
stop=time.time()

t3.append(stop - start)


start=time.time()
mergeSort4arrays(arr3)
stop=time.time()

t4.append(stop - start)
```

For array size 10000

```
arr=[]
arr1=[]
arr2=[]
```

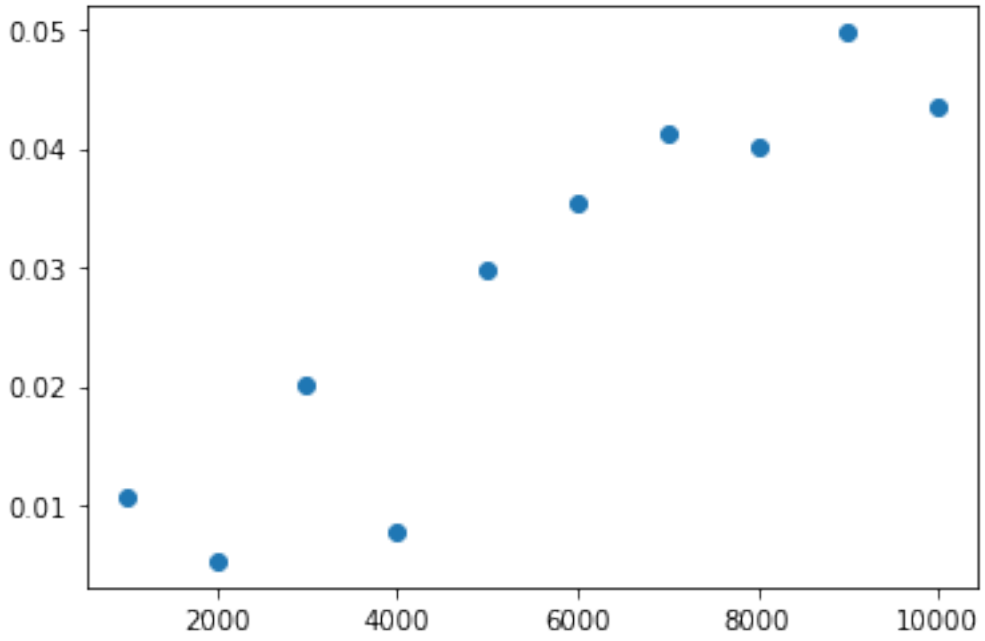
```
arr3=[]
for i in range(10000):
    a=rn.randint(0,100000)
    arr.append(a)

for i in range(10000):
    arr1.append(arr[i])
    arr2.append(arr[i])
    arr3.append(arr[i])

start=time.time()
mergeSort2arrays(arr1)
stop=time.time()

t2.append(stop - start)


start=time.time()
mergeSort3arrays(arr2)
stop=time.time()

t3.append(stop - start)


start=time.time()
mergeSort4arrays(arr3)
stop=time.time()

t4.append(stop - start)
```

Scatterplot of execution time vs array size

Splitting into 3

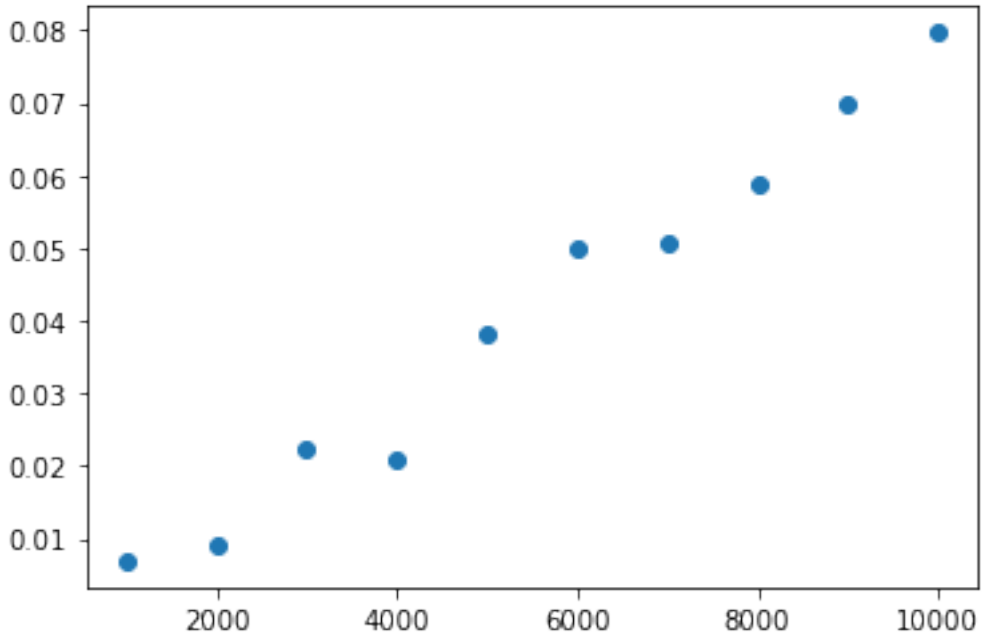
```
plt.scatter(n,t3)
plt.show
```

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




```
plt.scatter(n,t4)
plt.show
```

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

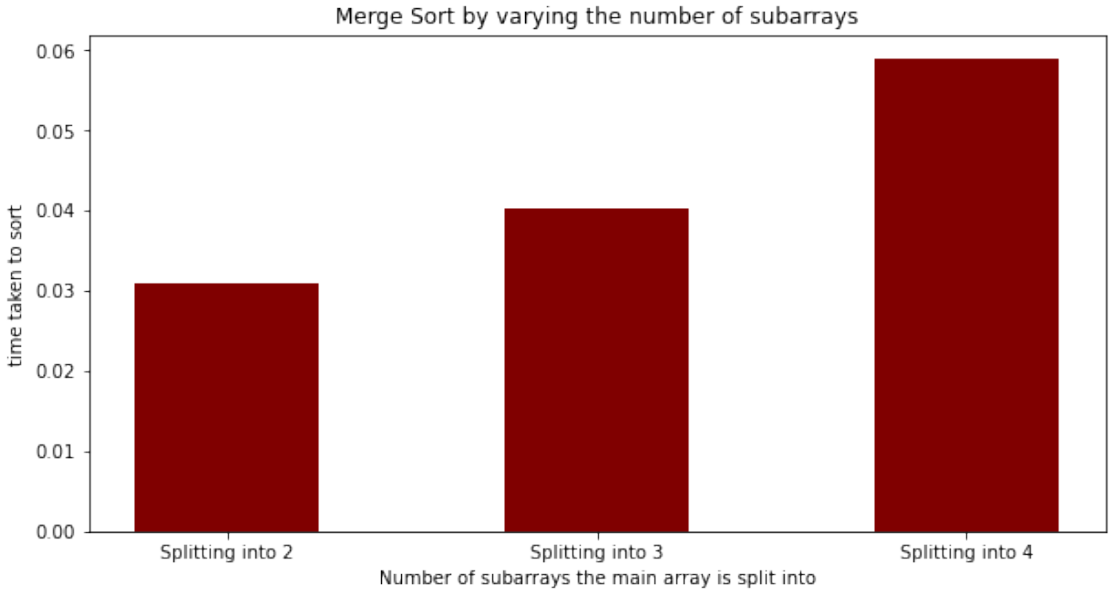



```
#print(t2[7],t3[7],t4[7])
```

```
0.030879497528076172 0.04020953178405762 0.05897188186645508
```

```
import numpy as np
import matplotlib.pyplot as plt
```

```
data = {'Splitting into 2': t2[7], 'Splitting into 3': t3[7],
'Splitting into 4':t4[7]}
subarraysize = list(data.keys())
time = list(data.values())

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


plt.bar(subarraysize, time, color ='maroon',
        width = 0.5)

plt.xlabel("Number of subarrays the main array is split into")
plt.ylabel("time taken to sort")
plt.title("Merge Sort by varying the number of subarrays")
plt.show()
```



As we can clearly see the program is most efficient when it is split into 2 subarrays, It is most efficient