In [1]:

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

In [2]:

```
def lomuto_part(arr, start, end):
    index = end
    key = arr[index]
    i = -1
    for j in range(0,index):
        if arr[j] <= key:
            i = i + 1
            arr[i], arr[j] = arr[j], arr[i]

    arr[i + 1], arr[index] = arr[index], arr[i + 1]

    return i+1
```

In [3]:

```
def Quick_sort(A, start, end):
    if start < end:
        pivot = lomuto_part(A, start, end)
        Quick_sort(A, start , pivot-1 )
        Quick_sort (A, pivot + 1, end)
```

In [4]:

```
A = [2, 4, 3, 3 , 1, 0, 10]
n = len(A)
Quick_sort(A, 0, n-1)
print(A)
```

```
[0, 1, 2, 3, 3, 4, 10]
```

In [5]:

```
Best_case = [0, 1, 3, 2, 6, 5, 9, 7, 10, 4]
length = len(Best_case)

n = length - 1
st = time.time()
Quick_sort(Best_case, 0 , n)
et = time.time()
elapsed_time = et - st
print('Best Case Execution time:', elapsed_time, 'seconds')
```

```
Best Case Execution time: 3.409385681152344e-05 seconds
```

In [6]:

```
worst_case = [10, 9, 8 , 7, 6, 4, 2, 1, 0]
length = len(worst_case)
n = length - 1

st = time.time()
Quick_sort(worst_case, 0 , n)
et = time.time()
elapsed_time = et - st
print('Worst Case Execution time:', elapsed_time, 'seconds')
```

```
Worst Case Execution time: 3.6716461181640625e-05 seconds
```

In [7]:

```
avg_case = [0, 1, 2 , 7, 6, 10, 9 ,  8 , 4]
length = len(avg_case)
n = length - 1
```

```
st = time.time()
Quick_sort(avg_case, 0 , n)
et = time.time()
elapsed_time = et - st
print('Avg Case Execution time:', elapsed_time, 'seconds')
```

```
Avg Case Execution time: 3.504753112792969e-05 seconds
```

In [8]:

```
input_list = [10, 1000, 5000, 10000]
time_taken = []
for i in input_list:
    random.seed(10)
    randomlist = random.sample(range(0 , i), i)
    length = len(randomlist)
    n = length - 1
    st = time.time()
    Quick_sort(randomlist, 0 , n)
    et = time.time()
    elapsed_time = et - st
    time_taken.append(elapsed_time)
    print('When input is', i ,': Execution time:', elapsed_time, 'seconds')
```

```
When input is 10 : Execution time: 9.059906005859375e-06 seconds
When input is 1000 : Execution time: 0.044013023376464844 seconds
When input is 5000 : Execution time: 1.1034939289093018 seconds
When input is 10000 : Execution time: 4.4688661098480225 seconds
```

plt.figure(figsize=(10,20)) plt.plot(input_list,time_taken, 'ro') plt.xticks(input_list) plt.yticks(time_taken) plt.xlabel("input size") plt.ylabel("time taken for sorting") plt.savefig('quick_sort_lomuto.png') plt.show() plt.close()