

**CENG313**  
**PROGRAMMING ASSIGNMENT 4 REPORT**  
Elif Yilmaz 230201054

## Sample Executions

### Executions with threads:

1 thread / 5 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 1 5
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Thread 0: task 0: 33 cannot be deleted
Thread 0: task 1: 27 cannot be deleted
Thread 0: task 2: 43 cannot be deleted
Thread 0: task 3: 36 is inserted
Thread 0: task 4: 49 cannot be deleted
main: Final list:
 36
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

1 thread / 7 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 1 7
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
Thread 0: task 0: 33 cannot be deleted
Thread 0: task 1: 27 cannot be deleted
Thread 0: task 2: 43 cannot be deleted
Thread 0: task 3: 36 is inserted
Thread 0: task 4: 49 cannot be deleted
Thread 0: task 5: 12 cannot be deleted
Thread 0: task 6: 40 cannot be deleted
main: Final list:
 36
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

1 thread / 10 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 1 10
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Thread 0: task 0: 33 cannot be deleted
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
Main: enqueued task 7: Delete 13
Main: enqueued task 8: Delete 39
Main: enqueued task 9: Delete 22
Thread 0: task 1: 27 cannot be deleted
Thread 0: task 2: 43 cannot be deleted
Thread 0: task 3: 36 is inserted
Thread 0: task 4: 49 cannot be deleted
Thread 0: task 5: 12 cannot be deleted
Thread 0: task 6: 40 cannot be deleted
Thread 0: task 7: 13 cannot be deleted
Thread 0: task 8: 40 is inserted
Thread 0: task 9: 22 cannot be deleted
main: Final list:
 36 40
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

2 threads / 5 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 2 5
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Thread 1: task 0: 33 cannot be deleted
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Thread 0: task 2: 43 cannot be deleted
Thread 0: task 3: 36 is inserted
Thread 0: task 4: 49 cannot be deleted
Thread 1: task 1: 27 cannot be deleted
main: Final list:
 36
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

## 2 threads / 7 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 2 7
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Thread 1: task 0: 33 cannot be deleted
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
Main: enqueued task 7: Insert 49
Main: enqueued task 8: Delete 12
Main: enqueued task 9: Delete 40
Thread 0: task 1: 27 cannot be deleted
Thread 0: task 3: 36 is inserted
Main: enqueued task 9: 49 cannot be deleted
Thread 0: task 5: 12 cannot be deleted
Thread 0: task 6: 40 cannot be deleted
Thread 1: task 2: 43 cannot be deleted
main: Final list:
36
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$
```

## 2 threads / 10 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 2 10
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
Main: enqueued task 7: Delete 13
Main: enqueued task 8: Insert 40
Main: enqueued task 9: Delete 22
Thread 0: task 0: 33 cannot be deleted
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
Main: enqueued task 7: Delete 13
Main: enqueued task 8: Insert 40
Main: enqueued task 9: Delete 22
Thread 0: task 1: 27 cannot be deleted
Thread 0: task 2: 43 cannot be deleted
Thread 0: task 3: 36 is inserted
Thread 0: task 4: 49 cannot be deleted
Thread 0: task 5: 12 cannot be deleted
Thread 0: task 6: 40 is inserted
Thread 0: task 7: 13 cannot be deleted
Thread 0: task 8: 40 is inserted
Thread 0: task 9: 22 cannot be deleted
Thread 1: task 5: 12 cannot be deleted
main: Final list:
36
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$
```

## 4 threads / 5 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 4 5
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Thread 1: task 0: 33 cannot be deleted
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Thread 1: task 3: 36 is inserted
Thread 0: task 2: 43 cannot be deleted
Thread 2: task 4: 49 cannot be deleted
main: Final list:
36
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$
```

## 4 threads / 7 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 4 7
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Thread 1: task 0: 33 cannot be deleted
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Thread 2: task 1: 27 cannot be deleted
Thread 2: task 2: 43 cannot be deleted
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
Thread 2: task 3: 36 is inserted
Thread 2: task 4: 49 cannot be deleted
Thread 3: task 5: 12 cannot be deleted
Thread 3: task 6: 40 cannot be deleted
Thread 3: task 7: 40 cannot be deleted
main: Final list:
36
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$
```

## 4 threads / 10 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 4 10
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
Main: enqueued task 7: Insert 39
Main: enqueued task 8: Insert 40
Main: enqueued task 9: Delete 22
Thread 3: task 3: 36 is inserted
Thread 3: task 5: 12 cannot be deleted
Thread 1: task 4: 49 cannot be deleted
Thread 2: task 6: 40 cannot be deleted
Thread 0: task 7: 13 cannot be deleted
Thread 3: task 9: 22 cannot be deleted
main: Final list:
36 40
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$
```

## 8 threads / 5 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 8 5
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Thread 2: task 0: 33 cannot be deleted
Thread 5: task 1: 27 cannot be deleted
Thread 2: task 2: 43 cannot be deleted
Thread 3: task 3: 36 is inserted
Thread 0: task 4: 49 cannot be deleted
main: Final list:
    36
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

## 8 threads / 7 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 8 7
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
Thread 5: task 0: 33 cannot be deleted
Thread 4: task 1: 27 cannot be deleted
Thread 2: task 2: 43 cannot be deleted
Thread 4: task 3: 36 is inserted
Thread 1: task 4: 49 cannot be deleted
Thread 3: task 5: 12 cannot be deleted
Thread 5: task 6: 40 cannot be deleted
main: Final list:
    39
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

## 8 threads / 10 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 8 10
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Thread 0: task 0: 33 cannot be deleted
Thread 0: task 1: 27 cannot be deleted
Main: enqueued task 4: Delete 49
Thread 3: task 2: 43 cannot be deleted
Thread 2: task 3: 36 is inserted
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Insert 40
Thread 1: task 4: 49 cannot be deleted
Main: enqueued task 7: Delete 13
Main: enqueued task 8: Insert 40
Main: enqueued task 9: Delete 22
Thread 5: task 5: 12 cannot be deleted
Thread 2: task 7: 13 cannot be deleted
Thread 4: task 8: 40 cannot be deleted
Thread 4: task 9: 22 cannot be deleted
Thread 7: task 8: 40 is inserted
main: Final list:
    36 40
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

## 16 threads / 5 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 16 5
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Thread 1: task 0: 33 cannot be deleted
Thread 4: task 1: 27 cannot be deleted
Thread 6: task 2: 43 cannot be deleted
Thread 5: task 3: 36 is inserted
Thread 10: task 4: 49 cannot be deleted
main: Final list:
    36
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

## 16 threads / 7 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 16 7
Hello from main thread.
Main thread start thread creation.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Thread 2: task 0: 33 cannot be deleted
Thread 0: task 1: 27 cannot be deleted
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
Thread 4: task 2: 43 cannot be deleted
Thread 13: task 6: 40 cannot be deleted
Thread 1: task 3: 36 is inserted
Thread 3: task 4: 49 cannot be deleted
Thread 2: task 5: 12 cannot be deleted
main: Final list:
    36
Main thread waits for threads to complete.
Threads are completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

## 16 threads / 10 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qt.out 16 10
Hello from main thread.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Thread 1: task 0: 0 can't be deleted
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Thread 5: task 1: 27 cannot be deleted
Thread 3: task 2: 43 cannot be deleted
Thread 2: task 3: 36 is inserted
Thread 4: task 4: 49 cannot be deleted
Main: enqueued task 6: Delete 40
Main: enqueued task 7: Delete 13
Main: enqueued task 8: Insert 40
Thread 6: task 5: 12 cannot be deleted
Thread 7: task 6: 40 cannot be deleted
Thread 8: task 7: 13 cannot be deleted
Thread 9: task 8: 40 is inserted
Main: enqueued task 9: Delete 22
Thread 4: task 9: 22 cannot be deleted
main: Final list:
36 40
Main thread waits for threads to complete.
Main thread completed.
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

## Sequential executions:

### 5 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qs.out 5
Hello from main thread.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
task 0: 33 cannot be deleted
task 1: 27 cannot be deleted
task 2: 43 cannot be deleted
task 3: 36 is inserted
task 4: 49 cannot be deleted
main: Final list:
36
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

### 7 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qs.out 7
Hello from main thread.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
task 0: 33 cannot be deleted
task 1: 27 cannot be deleted
task 2: 43 cannot be deleted
task 3: 36 is inserted
task 4: 49 cannot be deleted
task 5: 12 cannot be deleted
task 6: 40 cannot be deleted
main: Final list:
36
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

### 10 tasks

```
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ ./qs.out 10
Hello from main thread.
Main thread start task creation.
Main: enqueued task 0: Delete 33
Main: enqueued task 1: Delete 27
Main: enqueued task 2: Delete 43
Main: enqueued task 3: Insert 36
Main: enqueued task 4: Delete 49
Main: enqueued task 5: Delete 12
Main: enqueued task 6: Delete 40
Main: enqueued task 7: Insert 13
Main: enqueued task 8: Insert 40
Main: enqueued task 9: Delete 22
task 0: 33 cannot be deleted
task 1: 27 cannot be deleted
task 2: 43 cannot be deleted
task 3: 36 is inserted
task 4: 49 cannot be deleted
task 5: 12 cannot be deleted
task 6: 40 cannot be deleted
task 7: 13 cannot be deleted
task 8: 40 is inserted
task 9: 22 cannot be deleted
main: Final list:
36 40
tardis@tardis-UX303UB:~/OS_2018_fall/HWs$ 
```

## Implementation Details

As is seen from the screen shots of sample executions, main thread generates tasks and enqueue them in the task queue.

Main thread also creates worker threads and worker threads in condition sleep at first. Every time main thread enqueue a task, it weaken a worker thread with a condition signal. Worker thread that finishes its task, return condition wait. When main thread enqueues all task it awakens all the worker threads with a condition broadcast. After the broadcast, threads do their jobs and do not return condition wait, instead they finishes Thread\_work() , main thread waits for worker threads and it print final list. To do this I use mutexes, condition variables and Boolean variables. They are :

**cond\_queue**, a condition variable

**queue\_mutex**, a mutex

**flag**, a Boolean variable for broadcast

I use them for deciding put to sleep or awake worker threads

**mutex**, a mutex

I use it for dequeue and enqueue. They should not be done at the same time.

**cond\_list**, a condition variable

**list\_mutex**, a mutex

**isOver**, a shared integer variable.

I use them to print final list. All task must be executed before printing final list.

**work\_mutex**, a mutex

I use it for task execution. It is a critical section, more than one task can not be executed at the same time.

### 1000 tasks execution :

sequential → Total time spent: 0.002810 sec

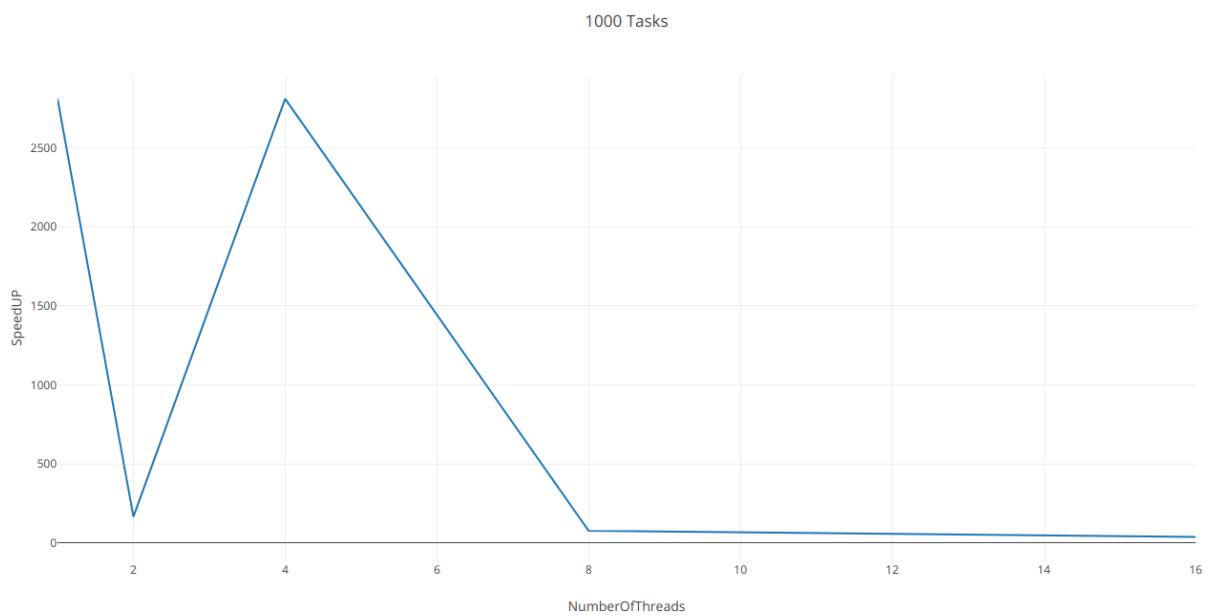
1 thread → Total time spent: 0.000001 sec  
speedup =  $0.002810 / 0.000001 = 2810$

2 threads → Total time spent: 0.000017 sec  
speedup =  $0.002810 / 0.000017 = 165.294$

4 threads → Total time spent: 0.000001 sec  
speedup =  $0.002810 / 0.000001 = 2810$

8 threads → Total time spent: 0.000037 sec  
speedup =  $0.002810 / 0.000037 = 75.945$

16 threads → Total time spent: 0.000076 sec  
speedup =  $0.002810 / 0.000076 = 36.973$



(Speedup =  $T_{\text{sequential}} / T_{\text{parallel}}$ )

### 10000 tasks execution :

sequential → Total time spent: 0.055274 sec

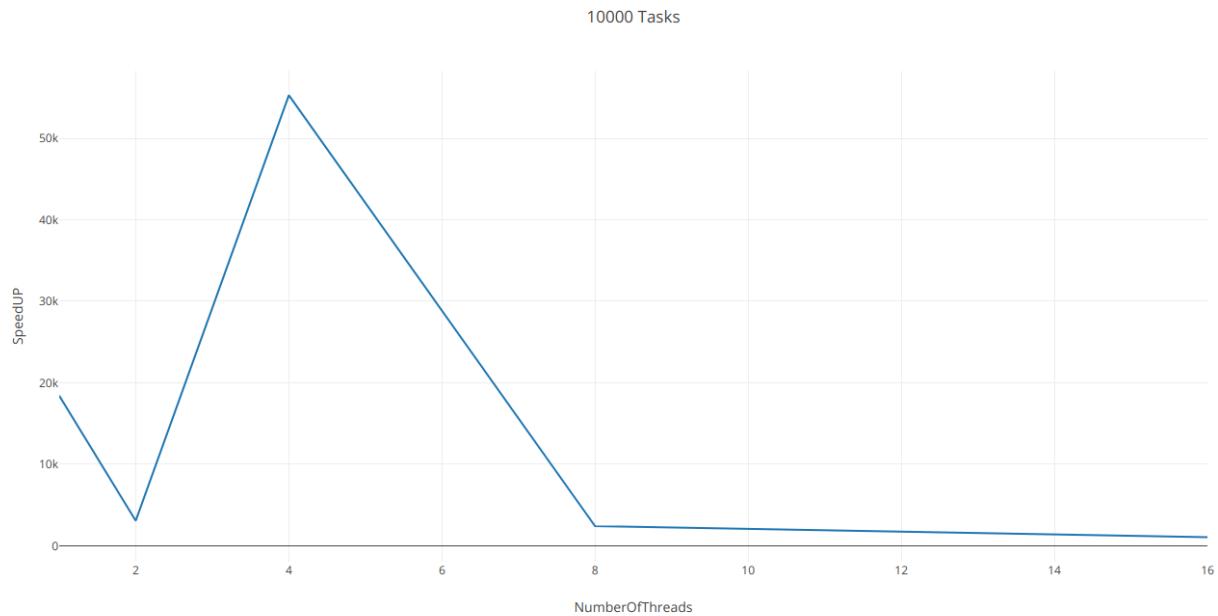
1 thread → Total time spent: 0.000003 sec  
speedup =  $0.055274 / 0.000003 = 18424.666$

2 threads → Total time spent: 0.000018 sec  
speedup =  $0.055274 / 0.000018 = 3070.777$

4 threads → Total time spent: 0.000001 sec  
speedup =  $0.055274 / 0.000001 = 55274$

8 threads → Total time spent: 0.000023 sec  
speedup =  $0.055274 / 0.000023 = 2403.217$

16 threads → Total time spent: 0.000053 sec  
speedup =  $0.055274 / 0.000053 = 1042.905$



(Speedup =  $T_{\text{sequential}} / T_{\text{parallel}}$ )

### 10000 tasks execution :

sequential → Total time spent: 0.487962 sec

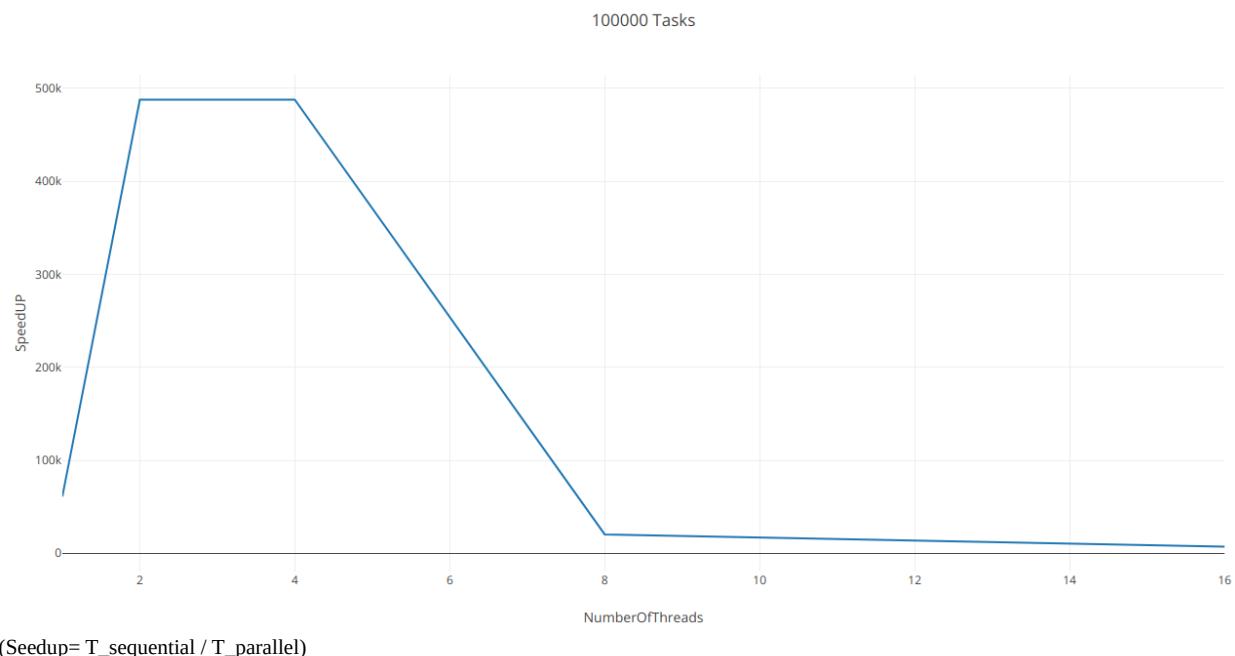
1 thread → Total time spent: 0.000008 sec  
speedup =  $0.487962 / 0.000008 = 60995.250$

2 threads → Total time spent: 0.000001 sec  
speedup =  $0.487962 / 0.000001 = 487962$

4 threads → Total time spent: 0.000001 sec  
speedup =  $0.487962 / 0.000001 = 487962$

8 threads → Total time spent: 0.000024 sec  
speedup =  $0.487962 / 0.000024 = 20331.750$

16 threads → Total time spent: 0.000069 sec  
speedup =  $0.487962 / 0.000069 = 7071.913$



## **Expectations and Observations**

Before executing the program with several threads, I expect that -if I code properly- computation time would decrease as number of threads increases. But I thought at some point with bigger number of threads program could slow down because of lockings and context switchings.

And I thought that there must be an optimal number of thread that runs the program fastest. I know that my machine has 2 cores and 2 threads per core. I assume that 4 thread would be optimal.

When I run the program and calculate speed ups, I see that 4 threads give me optimal results. In general, the program with threads run faster than the sequential program. I see the benefits of multi-threading most in the execution with 100000 task. Its speed up ratios are much bigger than the executions with 1000 tasks and executions with 10000 tasks.

Another interesting thing I observe that, in the executions with 1000 tasks and executions with 10000 tasks , there is a decrease of speed up as going from 1 thread to 2 thread.