

## TaskSet Generation Scheme Description

### Generate utilization vector

1. Given total utilization for the entire task set and  $N$  number of tasks, we randomly select  $(N - 1)$  values between 0 and  $U$ . *This procedure is  $O(N)$ .*
2. We add 0 and  $U$  to the series and sort the series with merge sort. *This step is  $O(N \log N)$*
3. For each value in the series, we take the difference between the next value and the current value. *This step is  $O(N)$ .*

The sum of those values is equivalent to the total utilization. The first set of values were randomly selected with the `rand()` function; Therefore, the difference of the values that make up the utilization vector in the taskSet are randomly distributed.

### Generate period vector

1. We randomly generate a real value from  $[1, 6]$  and take the base 10 power of this value to generate a random integer value.
2. We repeat the previous step  $N$  times to create a set of  $N$  random integer values for the period vector. *This step is  $O(N)$*
3. We apply type cast to convert the double value to integer for the values in the period vector.

### Generate execution time

1. For each item in utilization vector and period vector, we take the product of each item to generate the execution time for each task.

### Printing Task Set

1. Finally, we iterate through the utilization vector and period vector and print out the task set of values to a file. *This step is  $O(N)$ .*

The biggest time complexity in this algorithm is  $O(N \log N)$ . Hence, our program is  $O(N \log N)$ .