1)

I wrote a simple java code, and I found the value of n.



When n equals to 15, an algorithm whose running time is 100 is faster than algorithm whose running time is on same machine.

2)

We will try to maximize n value. To find, we should calculate how many microseconds makes 1 second, 1 minute etc.

First, I will find n values for . We know that if ,then . In that case,

Here, I will find values of n for . We know that if ,then .In that case,

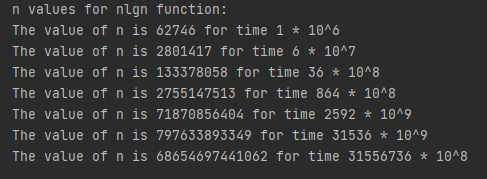
Here, I will find values of n for n. We will put time because we directly have n,

Here, I will find values of n for. We know that if ,then.In that case,

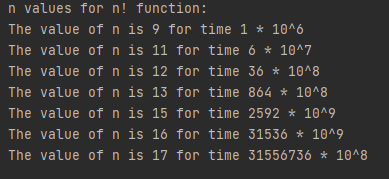
Here, I will find values of n for. We know that if ,then. These n values will be biggest number which holds equations. So,

Here, I will find values of n for. We know that if ,then. These n values will be biggest number which holds equations. So,

For nlogn, I wrote a java code which finds value of n. These are the results I get.



For n! I wrote a basic java code to find value of n.



So, the entire table becomes:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 second | 1 minute | 1 hour | 1 day | 1 month | 1 year | 1 century |
| lgn |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| n |  |  |  |  |  |  |  |
| n lgn | 62746 | 2801417 | 133378058 | 2755147513 | 71870856404 | 797633893349 | 68654697441062 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| n! | 9 | 11 | 12 | 13 | 15 | 16 | 17 |