1.How big of an array can I sort using merge-sort within one minute?

Merge sort runs in O(n log n) time because at each “level” of recursion it does an O(n) merge, and there are log n levels from repeatedly splitting the array in half.

The speed of modern processors is usually in the range between 1GHz and 4GHz, i.e., they run between 10^9 and 4 ⋅ 10^9 cycles per second. In a great simplification, a single “high level” operation – such as addition, multiplication, comparison, reading memory, etc. – takes several cycles depending on the type of operation. A ballpark estimate is that a program written in C or C++ would perform 10^8 such high level operations in a second. Python, being an interpreted language, would be about 10 times slower.

However, the concrete numbers – for running the same algorithm on the same input data – vary widely depending on software and hardware.

Therefore, if I run this code on Python, in one minute I am able to do 10^7 \* 60 high level operations, i.e 6 \* 10^8 , while in C or C++ I can do 6\* 10^9 high level operations

So the equation to solve for Python is roughly n \* log2(n) = 6 \* 10^8. This equation cannot be solved algebraically in a simple closed form, but we can approximate the solution numerically to n = 2.44 \* 10 ^ 7 and so for C or C++ it is n = 2.44 \* 10^8.  
  
In conclusion, it is important to underscore how these numbers can vary depending on the software(ex: Local or Cloud) and hardware(ex: GPU) used, so they should be tested empirically