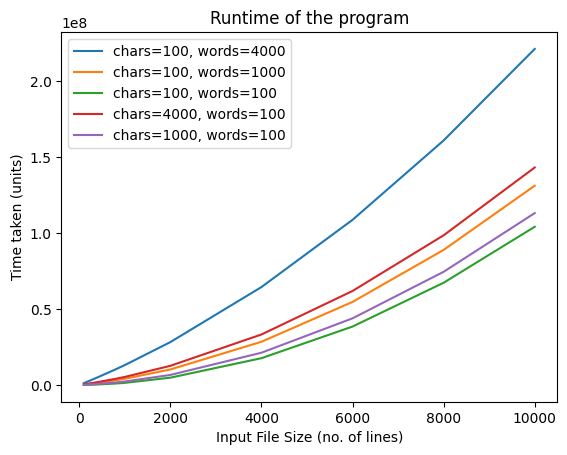
**GRAPH**



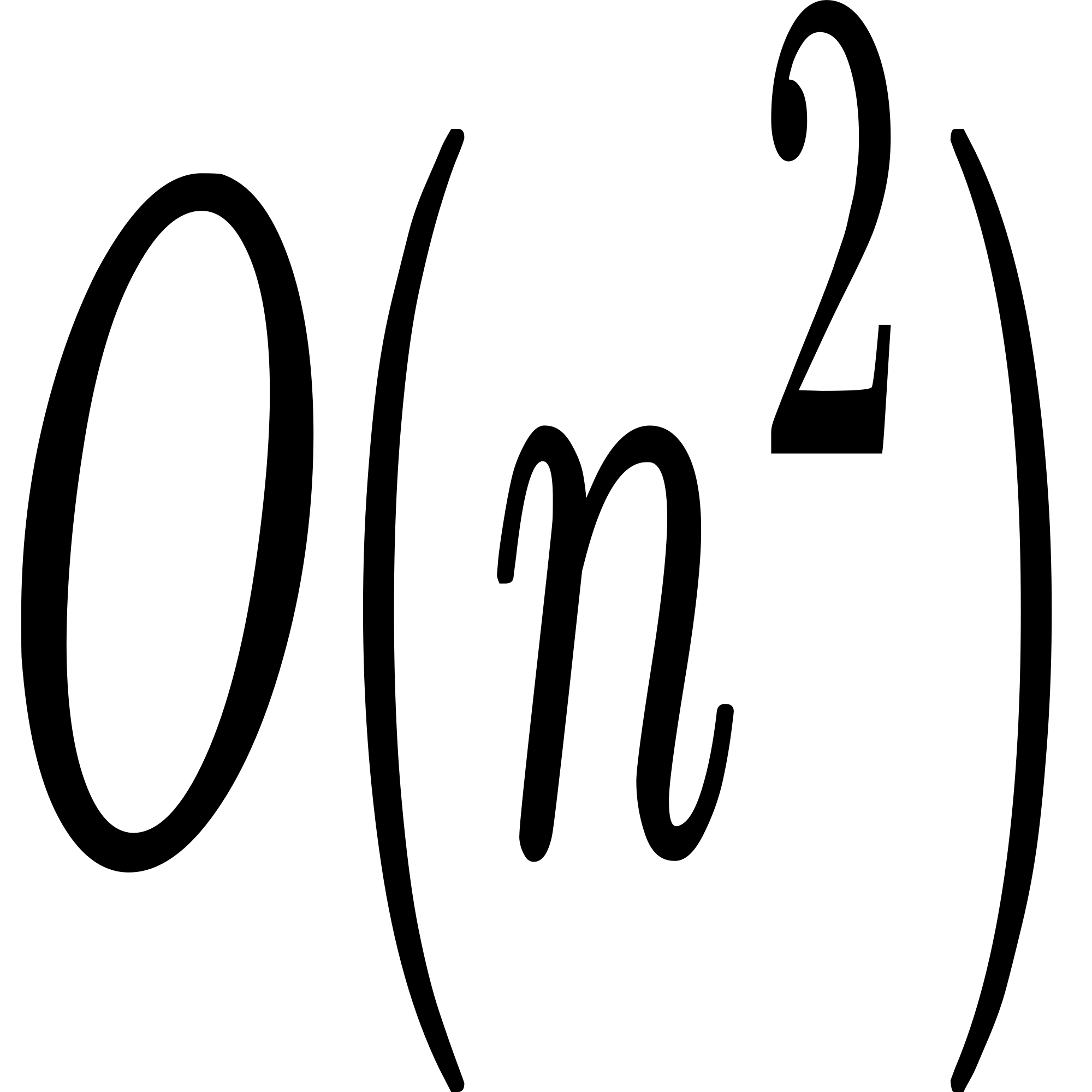
**EXPLANATION AND ANALYSIS**

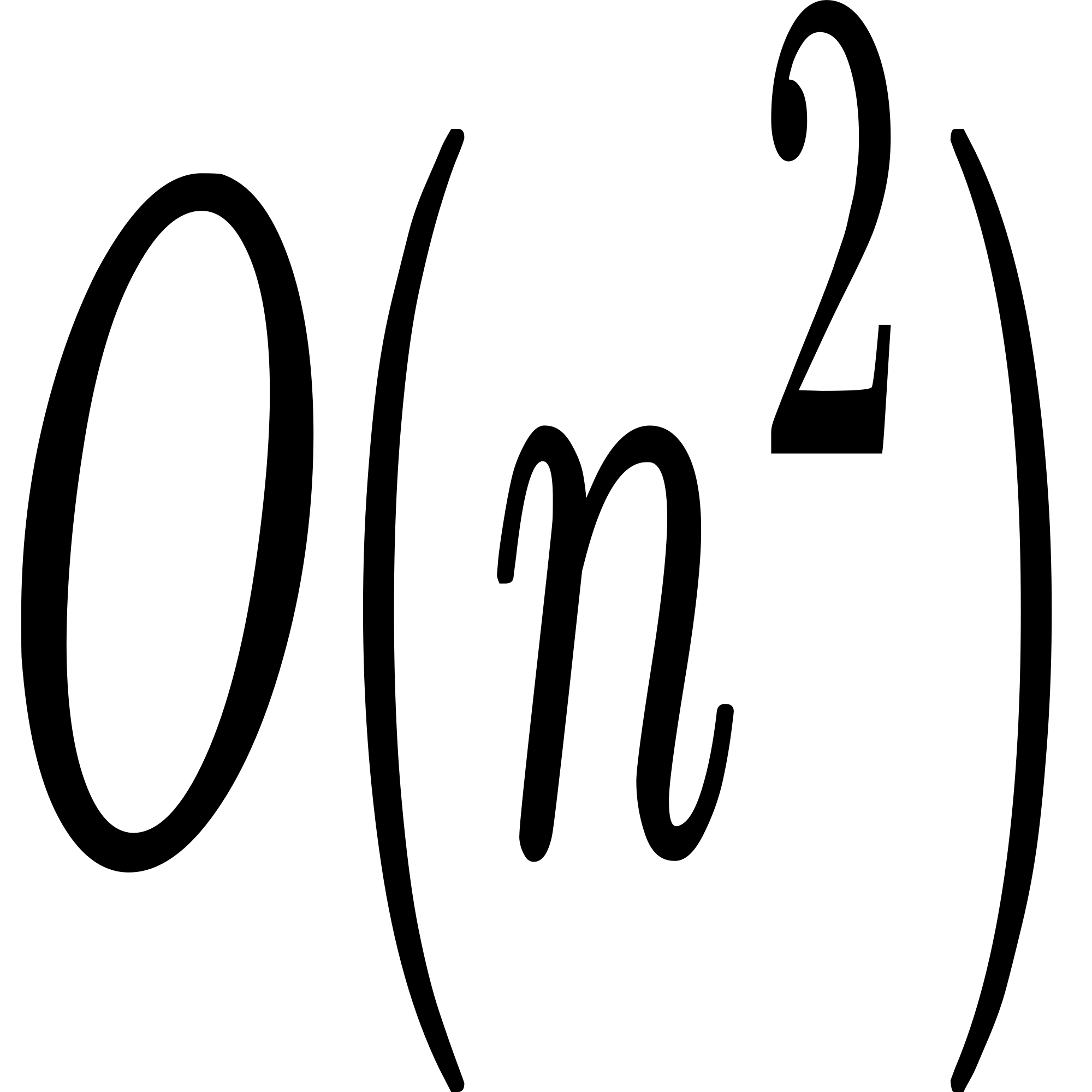
The terms <chars> and <words> in the diagram are explained below:

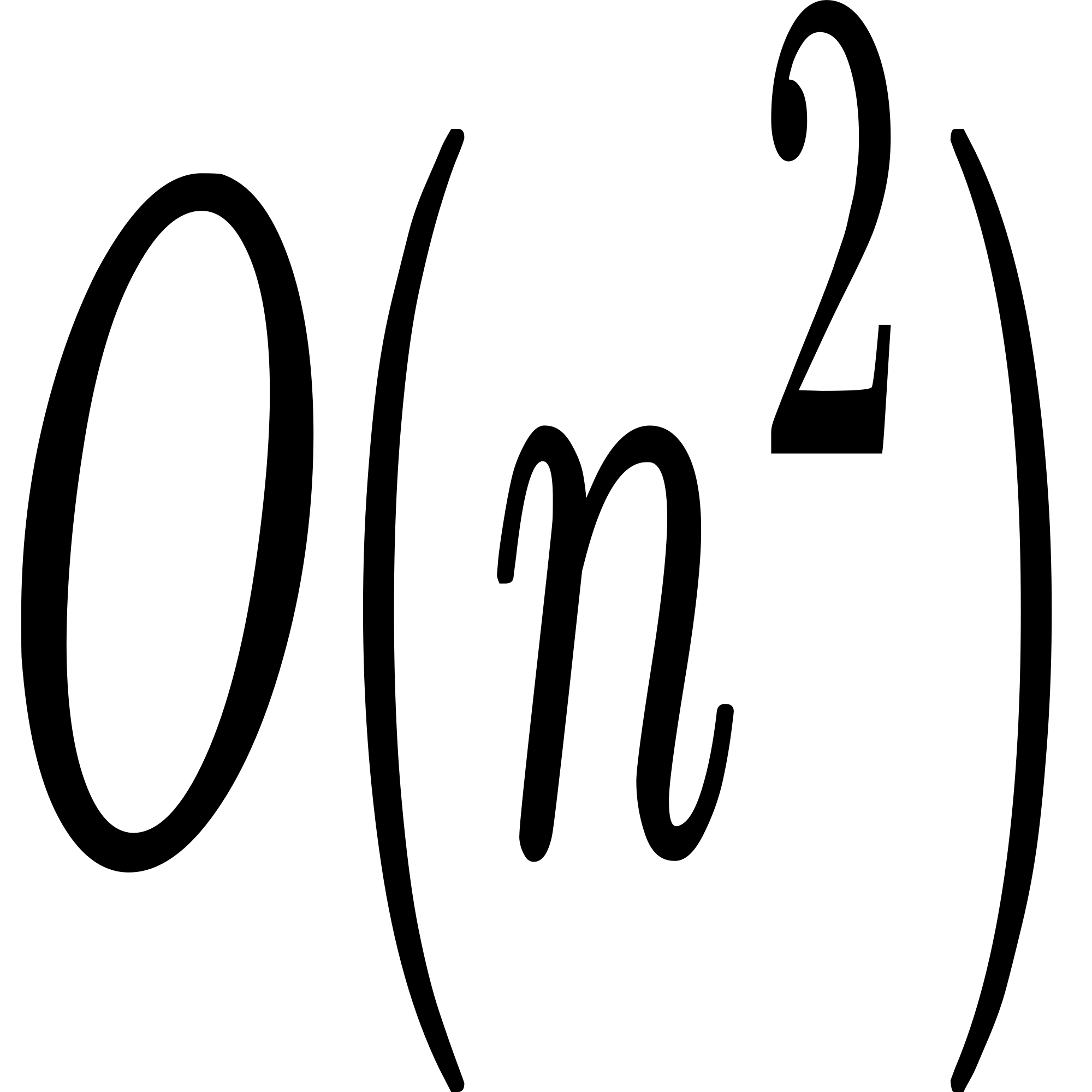
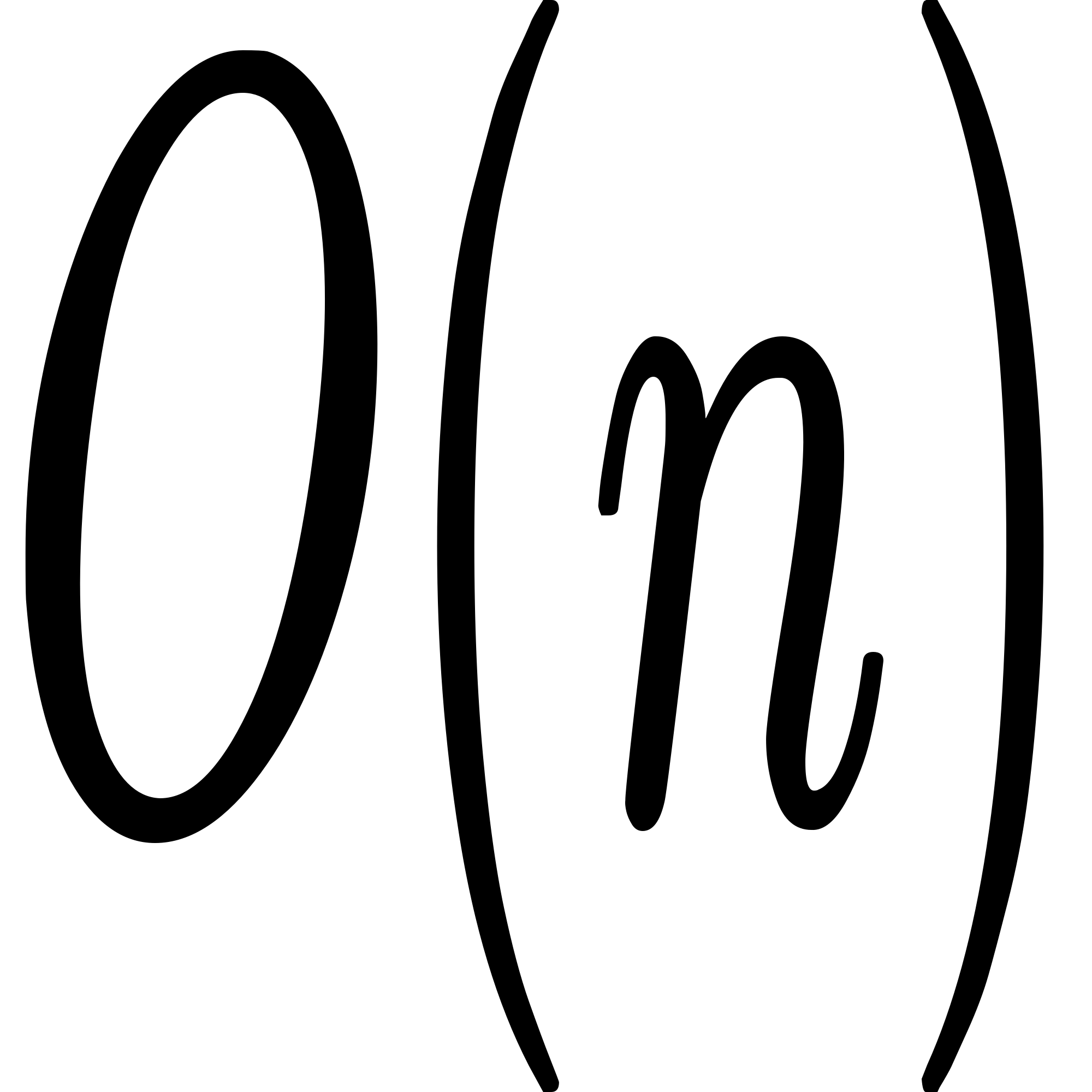
- chars stands for the number of characters in a word

- words represent the number of words that can be in a sentence or a line.

I factored using different values for both the words and the characters in a word because the each contribute to the running time of the program. If a text file being read from has longer sentences and/or longer words in a sentence, the running time would be affected by that, hence the choice of using a wider range of constants to represent the values (words & chars).

**-** The asymptotic time complexity of my program is and the shape of the graph suggests that quadratic relationship between the file input size and the running time of the program when given constant maximum values of ‘chars’ and ‘words’.

** COMPARING WITH O(n).**

****The time complexity is quadratic which implies that running time grows proportionally to the square of the input size (the number of lines in the text file). This means that if the input size is doubled, the running time quadruples. wps on the other hand implies that the running time grows proportionally to the input size, meaning that doubling the input size would nearly double the running time as well. The two time complexities are different, and comparing the quadrupling and doubling effect of running times based on doubling input size, performs better as the running time does not react as much as to file input size. Therefore, my program is not as efficient as O(n).