* big oh class of the sorting algorithm in question, with an explanation
  + The big O class for the bubble sort algorithm is O(n). By counting the number of loops, we see that only one while loop is present. This while loop runs in total relative to the input size.
* screenshots of the C Profiler
* graph plotting the size of the random arrays (x axis) and the time taken to sort (y axis)
* your reflections on how the C Profile data relates to what you have learned about algorithmic analysis and in particular to the big oh class of your algorithm.
  + It was interesting to see the difference between each array that was run through the bubble sort algorithm. The one in question has a time complexity of O(n), which is not bad. However, after calling bubble sort on arrays ranging from 1000 elements to 10,000, we proved that the execution time grows directly in proportion to the input size. The screenshots below show that the execution time for bubble sort grew from 0.051 seconds on 1,000 items to 4.751 seconds on 10,000 items.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.