We will do the ultimate smackdown of these two algorithms!

0.1 Time Complexity

0.1.1 Merge Sort

Merge sort uses recursion with $O(\log n)$ layers and $O(n/2^{\log l-1})$ splits per layer making the part where it splits into sets be $O(n)$. The merging process is $O(n+m)$ with n, m being the sizes of the lists to merge so it is the sum of all these - $O(n)$ per layer times $O(\log n)$ layers giving Merge Sort an $O(n \log n)$ time complexity!

0.1.2 Bubble Sort

Bubble Sort has an $O(n^2)$ time complexity from repeately bubbling up maximums.

0.1.3 Scoring

The win for time complexity goes to... MERGE SORT!
The score is $1-0$, Merge Sort

0.2 Memory Complexity

0.2.1 Merge Sort

Merge Sort uses $O(n)$ auxilary memory, as the merging process is $O(n+m)$ in time and memory.

0.2.2 Bubble Sort

Bubble Sort uses $O(1)$ when bubbling up.

0.2.3 Scoring

Bubble Sort gets a huge W here, making it tie game at $1-1$. It's a close race, and stability is the tiebreak!

0.3 Stability

0.3.1 Merge Sort

Every part including the merging process is stable, so Merge Sort is stable!

0.3.2 Bubble Sort

Bubble Sort is stable, as the process of ”bubbling up” is stable.

0.3.3 Scoring

It's tied, leaving us at $1.5 - 1.5$!

0.4 Tiebreaks and Algorithm Grades

Time complexity is the most important thing for a sorting algorithm, giving Merge Sort the tiebreak! It's so important that Merge Sort gets a **A-** due to memory complexity, but Bubble Sort being so slow gives it a **C+**.