**Platform Product Search**

Big O notation helps us understand how the efficiency of an algorithm changes with the size of the input. It measures how the time or space requirements of an algorithm grow as the input size increases. Here’s a breakdown of the common complexities:

* **O(1)**: Constant time – The algorithm takes the same amount of time no matter how large the input is.
* **O(log n)**: Logarithmic time – The time grows in proportion to the logarithm of the input size.
* **O(n)**: Linear time – The time grows directly in proportion to the input size.
* **O(n log n)**: Line arithemic time – The time grows proportionally to the input size times its logarithm.
* **O(n^2)**: Quadratic time – The time grows proportionally to the square of the input size.
* **O(2^n)**: Exponential time – The time doubles with each additional unit of input size.

**Best-Case Scenario**: The search is completed quickly and easily, with everything going smoothly and minimal risk or resource use.

**Average-Case Scenario**: The search takes a bit longer and requires more effort, but the missing person or item is found eventually. There are some obstacles, but they are manageable.

**Worst-Case Scenario**: The search is long and resource-intensive, with high risks involved. The outcome might be negative, or the missing person or item is found in poor condition. Poor planning and communication make the situation even more challenging.