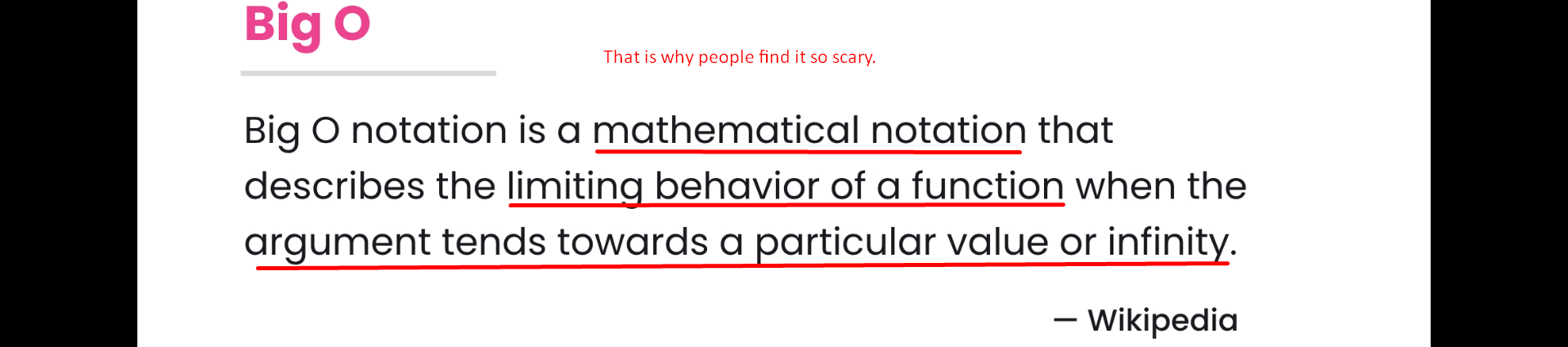
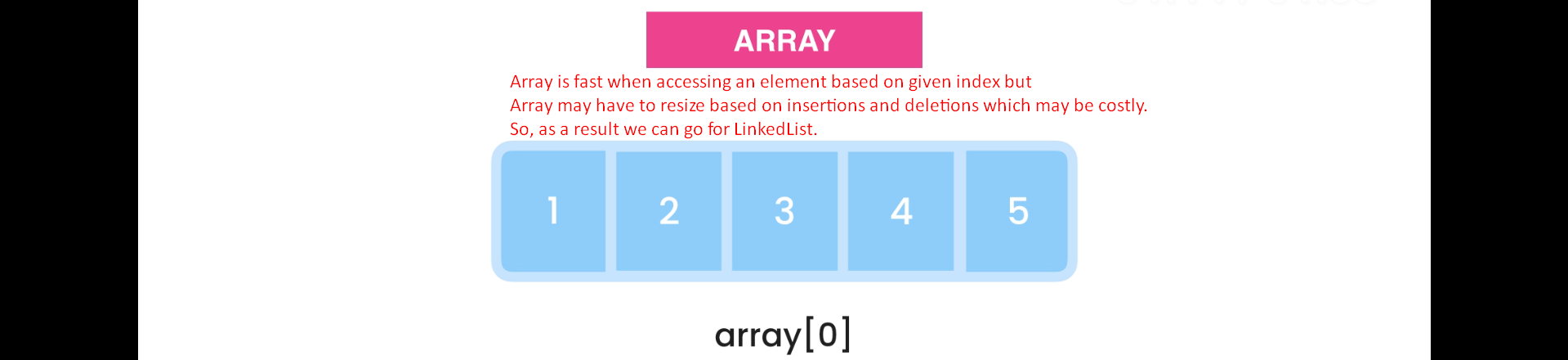
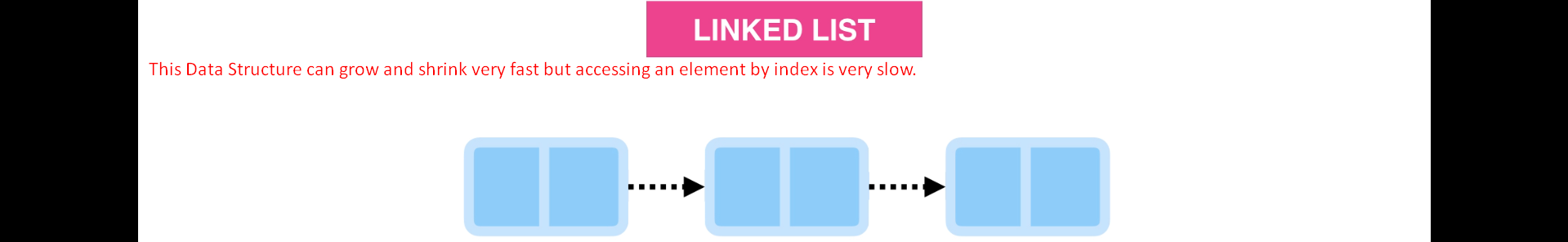
1. 
2. Let’s start with a classical def with Wikipedia.
3. 
4. 
   1. It helps us to determine whether the given algo is scalable or not which basically means whether the given algo is going to scale well when a very large input is given.
   2. Just because your code executes really quick and well doesn’t mean it’s going to perform well when a large dataset is given.
   3. So, that is why we have to use Big O to describe the performance of an algo.
   4. Jatin: Actually, measuring performance requires
      1. Algo (procedure)
      2. Underlying DS
      3. Operations and
      4. The input size. (We always take very big input size)
   5. What does it (Big O Notation) have to do with Data Structure?
      1. Some operations may be more or less costly based on what Data Structure we use.
      2. 
      3. 
      4. So, that is why you have to learn Big O notation first before you learn various Data Structures.  
         Also various companies such as Google, Amazon, Microsoft ask you about Big O.  
         They wanna know that you really know about how scalable a given algo is.  
         Moreover, knowing Big O will make you a better Developer or Software Engineer.
5. In next few lectures, we are going to use some code snippets to describe the Big O.