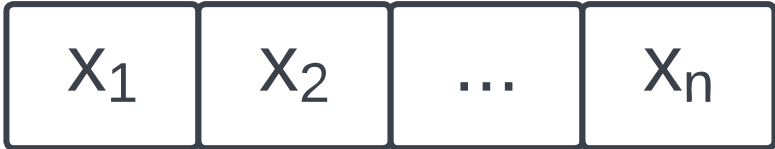


# Longest Consecutive Subsequence



Define Set:

$$A = \left\{ \begin{array}{l} \text{Set of all input} \\ \text{array values} \end{array} \right\}$$

A is created by copying contents of input array

A needs to be any data structure capable of  $O(1)$  lookup time for members

Ex: Python has a **Set** data structure that is implemented using a hash table where its keys and values are themselves members of the set



longestSequenceLength = 0



For each  $x_i$   
If predecessor $_{x_i} = (x_i - 1)$  **NOT** in Set A:

Then  $x_i$  is start of a sequence

sequenceLength =  $x_i + 1$

nextNum =  $x_i + 1$

While nextNum in Set A:

sequenceLength += 1

nextNum += 1

Build sequence while successors can be found

If sequenceLength > longestSequenceLength

then longestSequenceLength = sequenceLength

Once loop is done

Return longestSequenceLength