**Names : Aimee Nduwumwe (40086156)**
**Raghad Jaafar (40157929)**

```
  1)  Algorithm setThreshold(Threshold):
      Input The value separating when to use sequence or a more complex ADT
       Output : data structure
  CleverSIDC(int size)
          setSmartThresholdULS(size)
          if(isBST)
              s = new BStree()
              display the data structure used is Binary Search Tree

          else if(isSequence)
              s = new list()
              display the data structure used is List


  2)  Algorithms generate (n)
       Input how many keys to generate
      Output generates keys

do
          randomKey = 1000000 + r.nextInt(90000000)
              strVal = Integer.toString(randomKey)

          while(this.getValues(strVal) != null)
          String key=Integer.toString(randomKey)
          return key

      int num=0
      public String generatev()
          num++
          String v="testuser"+num
          return v
```

3) Algorithm allKeys():

Output all the keys inside the data structure

```
Node n = this.first
        keysToSort = new int[size]

        for(int i = 0; i<size; i++)
            if(n != null)
            keysToSort[i] = Integer.parseInt(n.getKey())
            n = n.next

        this.temp = new int[size]
        mergesort(0, size - 1)
        return keysToSort
```

4) Algorithms rangekey (key1 , key2)

Input: A list and two keys

Output: keys within a range

```
int numbOfKeys = 0
      String key = this.getFirstKey()

      while (key != null) {
          if( key.compareTo(key1) > 0 && key.compareTo(key2) < 0)
              numbOfKeys++
          key = this.nextKey(key)

      return numbOfKeys
```