Encoding:

- s = "WEEKEND"

- ```
   i     Original Suffixes
  --    ------------------
   0    W E E K E N D
   1    E E K E N D W
   2    E K E N D W E
   3    K E N D W E E
   4    E N D W E E K
   5    N D W E E K E
   6    D W E E K E N
  ```

- ```
   i     Original Suffixes      Sorted Suffixes
  --    ------------------      ---------------
   0    W E E K E N D           D W E E K E N
   1    E E K E N D W           E E K E N D W
   2    E K E N D W E           E K E N D W E
   3    K E N D W E E           E N D W E E K
   4    E N D W E E K           K E N D W E E
   5    N D W E E K E           N D W E E K E
   6    D W E E K E N           W E E K E N D
  ```

- t = , first =

Decoding

- t = , first =

- Sorted Suffixes

  ---------- N
  ---------- W
  ---------- E
  ---------- K
  ---------- E
  ---------- E
  ---------- D

- Sorted Suffixes

  D --------- N
  E --------- W
  E --------- E
  E --------- K
  K --------- E
  N --------- E
  W --------- D

- Sorted Suffixes next[]
  --- ---
  D --------- N 6
  E --------- W 2
  E --------- E 4
  E --------- K 5
  K --------- E 3
  N --------- E 0
  W --------- D 1

- Original Suffixes

  W --------- D
  E --------- W
  E --------- E
  K --------- E
  E --------- K
  N --------- E
  D --------- N

- s =

Some tips for implementation of decode:

- Given t and sortedT, we can find next[j] as follows: Let c be the jth character in sortedT. Let i be the earliest instance of c in sortedT that has not been used. Then next[j]=i. You must do this in linear, not quadratic time.
- Given next[j], we know that next[first] = 0th character of input string s. next[next[first]] = 1st character of input string s. next[next[next[first]]] = 2nd character, and so forth. This should be easy to do in Java.
- The hardest part of this assignment is just understanding the decode process. Make sure you really understand this before attempting your implementation. Our reference solution for the entire decode method is only about 10 lines, not counting whitespace, comments, and declarations.