# Efficient Algorithms: Suffix Array Construction

## BruteForce:

|  |
| --- |
| import re  def getAllWords():      all\_words\_cleaned=[]      with open('./TomSawyerCompleteGutenberg.txt', encoding='utf-8-sig') as f:          for line in f:              # split on space and line endings              splittet\_line=line.split()                wordIndex = 0              replacedSpecialChars = []              for wordIndex in range(len(splittet\_line)):                  # find all special chars and replace them with nothign                  newWord = re.sub("[^A-Za-z0-9]+", "", splittet\_line[wordIndex])                  # make all words lowercase                  newWord = newWord.lower()                  replacedSpecialChars.append(newWord)                  wordIndex+=1              if(len(replacedSpecialChars)>0):                  for word in replacedSpecialChars:                      all\_words\_cleaned.append(word)      return all\_words\_cleaned  ## generate suffix array  def createSuffixArr(all\_words\_cleaned):      suffix\_arr = []      #generate suffix arr      for word in all\_words\_cleaned:          charIndex = 0          for charIndex in range(len(word)):              suffix = word[charIndex:]              suffix +="$"              suffix\_arr.append(suffix)      return suffix\_arr  ##sort suffix array  import time  i = 0  runs = 5  allTimesTogehter = 0  while i < runs:      start\_time = time.time()      allWords = getAllWords()      suffix\_array = createSuffixArr(allWords)      suffix\_array.sort()      currentTime = time.time()-start\_time      allTimesTogehter += currentTime      print("--- %s seconds ----" % (currentTime))      i +=1  print("--- %s All Runs ----" % (allTimesTogehter/5)) |

## Durchsnittliche Laufzeit bei 5 Durchläufen bei BruteForce:

--- 0.35420002937316897 seconds All Runs ----