

CPSC 320: TUTORIAL 10

1. You are given a string of n characters, $T[1 \dots n]$, which looks like a book from which all spaces and punctuation have been removed. (For example, it might be “itwasthebestoftimes...”)
You want to break the text into a sequence of valid words since you hope that will reveal the original story. You have a dictionary that provides the Boolean function $\text{dict}()$:

$$\text{dict}(w) = \begin{cases} \text{true} & \text{if } w \text{ is a valid word} \\ \text{false} & \text{otherwise.} \end{cases}$$

Give a dynamic programming algorithm to determine if the string T can be broken into a sequence of valid words. Your algorithm should take time $O(n^2)$ assuming calls to $\text{dict}()$ take constant time.

How would you modify your algorithm to produce the corresponding sequence of words?

2. Given an unlimited number of coins with denominations x_1, x_2, \dots, x_n , and an amount A , find the minimum number of coins needed to make the amount A or output “impossible” if amount A cannot be made.

Your algorithm should run in time $O(nA)$.

Why doesn't greedy work?