In word segmentation, you are given as input a string of alphabetical characters ([a − z]) without whitespace, and your goal is to insert spaces into this string such that the result is the most fluent according to the language model.  
a. Suppose that we have a unigram model u and we are given the string breakfastservedinside. The  
unigram costs of words are given as u(break) = 3, u(fast) = 6, u(breakfast) = 8, u(served) = 8, u(in) = 3, u(side) = 5, u(inside) = 2. Assume u(s) = 100 for any other substring s of our string.

Consider the following greedy algorithm: begin at the front of the string. Find the ending position for  
the next word that minimizes the language model cost. Repeat, beginning at the end of this chosen  
segment.  
What is the total model cost from running this greedy algorithm on breakfastservedinside? Is this  
greedy search optimal for general inputs? In other words, does it find the lowest-cost segmentation of  
any input? Explain why or why not in 1-2 sentences.  
What we expect: The value of the total model cost and an explanation of why the greedy algorithm  
is or is not optimal. (Общая стоимость модели и объяснение почему жадный алгоритм оптимален или нет)

Your Solution:

Общая стоимость модели: breakfast + served + inside = 8 + 8 + 2 = 18

Нет, этот жадный алгоритм не является оптимальным для общих входных данных, так как он не учитывает следующие слова, которые могут иметь более высокую стоимость.

Возможный вариант - лучше использовать слово с более высокой стоимостью из двух вариантов, чтобы получить более низкую стоимость для следующего слова.

Пример

Пусть дана строка weekendsnow и стоимости

week = 3, end = 5, weekend = 1, snow = 15, endsnow = 2

Жадный алгоритм: weekend + snow = 1 + 15 = 16

Оптимальный: week + endsnow = 3 + 2 = 5