BLG372E (Analysis of Algorithms)
Spring 2012
Project # 2

Handed Out: 17.04.2012
Due: 30.04.2012 23.00

Word wrapping problem which is defined as breaking a long line into small and bounded line pieces can be solved using dynamic programming or greedy approach. In this project, word wrapping problem will be studied. Consider the following line,

i've heard there was a secret chord that david played, and it pleased the lord

If the given line above is divided into smaller lines with maximum 40 characters in each line, seperation could be performed as this.

i've heard there was a secret chord
that david played, and it
pleased the lord

However, seperation above is not optimal. It is stated that each line can have 40 characters. But the first line has 35 characters, the second line has 25 characters and finally the last line has 16 characters. Let's consider a cost metric which is defined as the third power of empty lines in each line except for the last line. Then the cost of the seperation above is

$(40 - 35)^3 + (40 - 25)^3 = 125 + 3375 = 3500$

According to given information above, answer the following questions using necessary implementations.

Question 1

(40 points)

Using C/C++, implement the following algorithm which is focused on word wrapping problem.

```
1: INPUT : text, maximum_char_per_line
2: OUTPUT : cost
3: line_length ← 0
4: cost = 0
5: for w is the next word of the text do
6:    if w_length + line_length ≤ maximum_char_per_line then
7:       line_length ← line_length + w_length + 1
8:    else
9:       cost = cost + (max_char_per_line − (line_length − 1))^3
10:      line_length ← w_length + 1
11:   end if
12: end for
13: return cost
```

What is the time complexity of this algorithm? Calculate the complexity and give the exact running times in a tableau. Run your implementation on test texts. Plot the following graphics (you can give these graphics as histograms),

- x axis represents the number of words in test text and y axis represents the exact running time.
- x axis represents the test text, y axis represents the cost.

For *max_char_per_line* value, you can select an acceptable value which is greater than the longest word length in the document (i think 40-50 is ok).

Question 2

(40 points)

According to dynamic programming aspects, design an algorithm to obtain line breaking with optimal cost. Implement your algorithm using C/C++ and state time complexity of it. Similarly with question 1, give cost and exact execution time of your program for each test text and plot graphics.

Question 3

(20 points)

Compare the resuls of question 1 and 2. Explain these results and their relations briefly. According to the comparison of results, which solution would you prefer in which conditions?

Academic Honesty Policy

You may discuss the problem addressed by the homework at an abstract level with your classmates, but you should not share or copy the solution from your classmates or from Internet. You should submit your own, individual homework. Plagiarism and any other forms of cheating will have serious consequences, including failing the course.

Submission Instruction

Submit your code(s) and report through Ninova. Please don't send any submission via e-mail. For your questions, you can send mail (ataka@itu.edu.tr) or see me at 4308 in weekdays except for Wednesdays.