University of Washington April 15, 1998
Department of Computer Science and Engineering
CSE 521, Spring 1998

Homework 3, **Due April 22, 1998**

Problem 1. String alignment with gaps:

A cost model used in biology for string alignment is the "affine cost gap model," where a gap of length k is charged $\alpha + k$. With this cost model, the optimal alignment problem is equivalent to finding an alignment which minimizes:

$$\text{length} + \alpha \times \text{gaps}.$$

Give an $O(nm)$ algorithm for finding the optimal alignment of a pair of strings of length n and m respectively using the affine cost model.

Problem 2. Subset sum:

The subset sum problem is: given a set $S = \{s_1, \ldots, s_n\}$ of integers, and an integer K, determine if there exists a subset $S' \subseteq S$ which sums to exactly K.

Show that the subset sum problem can be solved in $O(nK)$ time using $O(K)$ space (including finding the subset S' if it exists).

Can you improve your solution to $O(K/\log n)$? (If this seems too trivial, you may be overlooking an important detail!)

Problem 3. Optimal typesetting:

Describe an algorithm for the optimal typesetting of a page of text.

- Text Model - The document consists of a collection of paragraphs and headings. You can assume a fixed font size for paragraph text, and different fixed font size for the headings.
- Page Model - The page can have multiple columns, and there may be areas of the page where text cannot be placed. For example, there may be some pictures already placed on the page. These can occupy just a portion of a column, so text will flow around them.
- Vertical Grid - (Important) Assume that there is a vertical grid. This means that every row of text or heading has its baseline on a row of the grid. (For example, the text font height might be 12 units, the heading font height 20 units, and grid spacing 3 units. In this case, an extra unit of space would need to be inserted between a heading and paragraph text to make things line up.) Let R be the number of grid rows in a column. Your algorithm's runtime may depend upon R.

- Vertical Spacing - There is a certain amount of space which should be left between headings and text, and also a certain amount of space between paragraphs. If necessary, you can leave extra space (although a penalty will be charged).
- Optimization Problem - Penalties are assessed for poor line breaks and for using extra white space. Other typographic no-no's include headings at the bottom of a column, and widows and orphans. (I believe a widow is a line or two of isolated text at the bottom of a column, and an orphan is a line or two of isolated text at the top of a column, but it could be the other way round.) Give an algorithm which finds a typographical layout which minimizes the total penalty. Your algorithm should be polynomial in R and n where n is the number of words.
- Common Sense - use your common sense in figuring out other details of this problem. You will need to specify a number of the parameters of this problem. Give a high level description of your algorithm, and describe how it handles line-breaks, column breaks (to avoid widows, orphans, isolated headings), flowing text around pictures, and adding extra white space.

Problem 4. From the text:

Page 327, problem 16-5. Note that there can be several edges with the same label leaving a vertex.

Problem 5. Solve problem 5a or 5b:

I would like to assign problem 5a, which is a relatively simple NP-completeness proof. However, I know there are a few students in class who haven't seen NP-completeness – so to be not totally unreasonable, you can do either 5a or 5b - correct solutions will be given the same weight.

Problem 5a. Vector subset sum:

The vector subset sum problem is: given a set $S = \{\vec{s_1}, \ldots, \vec{s_n}\}$ of m dimensional vectors of positive integers, and an m dimensional vector $\vec{b}$, does there exist a subset $S' \subseteq S$ such that $\sum_{s \in S'} s = \vec{b}$.

If the maximum entry of $\vec{b}$ is bounded by B, the problem can be solved in $O(nB^m)$, so if m is viewed as a constant and B is polynomial in n, the problem can be solved in polynomial time. (This result will be shown in class).

Prove that the problem is NP-complete when m is part of the input, and the maximum entry in $\vec{b}$ is bounded by n. I would prefer you to give a reduction from SATISFIABILITY, but you can use other well known NP-complete problems if you prefer. (This result actually holds for the case that the vector $\vec{b}$ is the vector $(1, 1, \ldots, 1)$.)

Problem 5b. From the text:

Page 324, Problem 16-1.