Problem Set 8

MIT students: This problem set is due in lecture on *Wednesday, November 14*.

SMA students: This problem set is due after recitation on *Friday, November 16*.

Reading: Chapters 15, 16, 23

Both exercises and problems should be solved, but *only the problems* should be turned in. Exercises are intended to help you master the course material. Even though you should not turn in the exercise solutions, you are responsible for material covered by the exercises.

Mark the top of each sheet with your name, the course number, the problem number, your recitation instructor and time, the date, and the names of any students with whom you collaborated.

MIT students: Each problem should be done on a separate sheet (or sheets) of three-hole punched paper.

SMA students: Each problem should be done on a separate sheet (or sheets) of two-hole punched paper.

You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudocode.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Graders will be instructed to take off points for convoluted and obtuse descriptions.

Exercise 8-1. Do exercise 15.4-4 on page 356 of CLRS

Exercise 8-2. Do exercise 16.2-2 on page 384 of CLRS.

Exercise 8-3. Do exercise 16.2-3 on page 384 of CLRS.

Exercise 8-4. Do exercise 23.1-8 on page 567 of CLRS.

Exercise 8-5. Do exercise 23.2-2 on page 573 of CLRS.

Problem 8-1. Typesetting

In this problem you will write a program (real code that runs!) to solve the following typesetting scenario. **Because of the trouble you may encounter while programming, we advise you to START THIS PROBLEM AS SOON AS POSSIBLE.**

You have an input text consisting of a sequence of n words of lengths $\ell_1, \ell_2, \ldots, \ell_n$, where the length of a word is the number of characters it contains. Your printer can only print with its built-in Courier 10-point fixed-width font set that allows a maximum of M characters per line. (Assume that $\ell_i \leq M$ for all $i = 1, \ldots, n$.) When printing words i and $i+1$ on the same line, one space character (blank) must be printed between the two words. In addition, any remaining space at the end of the line is padded with blanks. Thus, if words i through j are printed on a line, the number of extra space characters at the end of the line (after word j) is

$$M - j + i - \sum_{k=i}^{j} \ell_k \, .$$

There are many ways to divide a paragraph into multiple lines. To produce nice-looking output, we want a division that fills each line as much as possible. A heuristic that has empirically shown itself to be effective is to charge a cost of the cube of the number of extra space characters at the end of each line. To avoid the unnecessary penalty for extra spaces on the last line, however, the cost of the last line is 0. In other words, the cost *linecost*(i, j) for printing words i through j on a line is given by

$$\textit{linecost}(i,j) = \begin{cases} \infty & \text{if words } i \text{ through } j \text{ do not fit on a line,} \\ 0 & \text{if } j = n \text{ (i.e., last line),} \\ \left(M - j + i - \sum_{k=i}^{j} \ell_k\right)^3 & \text{otherwise.} \end{cases}$$

The total cost for typesetting a paragraph is the sum of the costs of all lines in the paragraph. An optimal solution is an division of the n words into lines in such a way that the total cost is minimized.

(a) Argue that this problem exhibits optimal substructure.

(b) Recursively define the value of an optimal solution.

(c) Give an efficient algorithm to compute the cost of an optimal solution. Analyze the running time and storage space of your algorithm.

(d) Write code (in any programming language you wish[1]) to print an optimal division of the words into lines. For simplicity, assume that a ***word*** is any sequence of characters that are not blanks. Thus, a word is a substring strictly between two space characters or bounded by the beginning or end of the input.

[1]The solution will be written in C.

For part (d), you should turn in a printout of the code you have written and the output of your program on the input samples which are provided at the end of the problem set. Use two values of M (the maximum number of characters per line), namely $M = 72$ and $M = 40$, on each input sample. You can download these input samples from the 6.046 webpage (`http://theory.lcs.mit.edu/classes/6.046`). If, for some reason, you need to type the samples in yourself, please be careful about reproducing text accurately. For example, substituting double quotes in place of two single quotes will result in a different layout. Remember that collaboration, as usual, is allowed to solve problems, but you must write your program by yourself.

(e) Suppose instead that the cost of a line is defined as just the number of extra spaces. That is, when words i through j are put on a line, the cost of that line is

$$\mathit{linecost}'(i,j) = \begin{cases} \infty & \text{if words } i \text{ through } j \text{ do not fit on a line,} \\ 0 & \text{if } j = n \text{ (i.e., last line),} \\ M - j + i - \sum_{k=i}^{j} \ell_k & \text{otherwise;} \end{cases}$$

and that the total cost is still the sum of the costs of all lines in the paragraph. Give an efficient algorithm that finds an optimal solution in this case.

Problem 8-2. Scheduling to minimize average completion time

Suppose you are given a set S of n tasks, where task i requires p_i units of processing time to complete, once it has started. You have one computer on which to run these tasks, and the computer can run only one task at a time.

A ***schedule*** assigns which tasks run during what times on the computer. For any schedule, let c_i denote the ***completion time*** of task i, that is, the time at which task i completes processing. Your goal is to find a schedule that minimizes the average completion time, that is, minimizing

$$\frac{1}{n}\sum_{i=1}^{n} c_i \,.$$

(a) Suppose there are two tasks with $p_1 = 3$ and $p_2 = 5$. Consider (1) the schedule in which task 1 runs first, followed by task 2 and (2) the schedule in which task 2 runs first, followed by task 1. In each case, state the values of c_1 and c_2 and compute the average completion time.

(b) Give an algorithm that schedules the tasks so as to minimize the average completion time. Each task must run nonpreemptively, that is, once task i is started, it must run continuously for p_i units of time. Prove that your algorithm minimizes the average completion time, and state the running time of your algorithm.

Suppose now that the tasks are not all available at once. That is, each task has a ***release time*** r_i before which it is not available to be processed. Suppose also that we allow ***preemption***, so that a task can be suspended and restarted at a later time. For example, a task i with processing time

$p_i = 6$ may start running at time 1 and be preempted at time 4. It can then resume at time 10 but be preempted at time 11 and finally resume at time 13 and complete at time 15. Task i has run for a total of 6 time units, but its running time has been divided into three pieces.

(c) Give an algorithm that schedules the tasks so as to minimize the average completion time in this new scenario. Prove that your algorithm minimizes the average completion time, and state the running time of your algorithm.

Input sample 1. (From *A Capsule History of Typesetting*, by Brown, R.J.)
The first practical mechanized type casting machine was invented in 1884 by Ottmar Mergenthaler. His invention was called the "Linotype". It produced solid lines of text cast from rows of matrices. Each matrice was a block of metal – usually brass – into which an impression of a letter had been engraved or stamped. The line-composing operation was done by means of a keyboard similar to a typewriter. A later development in line composition was the "Teletypewriter". It was invented in 1913. This machine could be attached directly to a Linotype or similar machines to control composition by means of a perforated tape. The tape was punched on a separate keyboard unit. A tape-reader translated the punched code into electrical signals that could be sent by wire to tape-punching units in many cities simultaneously. The first major news event to make use of the Teletypewriter was World War I.

Input Sample 2. (From *an 1882 issue of the London Graphic.*)
The process of electrotyping may be briefly described as follows: The wood block or color plate is placed in a bed of wax, which has been melted, and allowed to cool until it has arrived at the proper consistency. It is then submitted to a great pressure in a press of hydraulic or other construction. In this way a facsimile of the original is produced but with every detail reversed. This was impression is then covered with a thin coating of black lead, such being a good conductor of electricity, and is hung by means of a brass rod in a large bath filled with a solution of sulfate of copper and sulfuric acid. Side by side with this bath is a powerful battery of Smee's construction, that is to say zinc, and platinized silver in dilute sulfuric acid. The current generated by this battery is put into connection with the wax mold hung in the bath, and also with a sheet of copper also hung side by side with the mold. The effect of the electricity is in the first place to decompose the copper and in the second place to attract the particles of copper to the mold. In a short time a thin coating of copper has formed along the mold, of which it is again the reverse, and consequently the exact facsimile of the original block. The shell, as it is called, is then filled up at the back with metal in order to make the surface perfectly hard and suitable for printing. After being made smooth and uniform in thickness by means of lathes and planing machines, it is mounted upon wood and is ready for the machine.

Input Sample 3. (Of *unknown* origin.)
The International Typographical Union, was described as, "the oldest union in America, and organized to prevent the use of labor saving improvements." The union fought hard for its members and when times were hard would send money and train fare to unemployed Typesetters, and direct

them to places where prospects were better. When preset advertising copy began to be provided by advertisers, in the late nineteenth century, the union required that this type could be used as received only if a union Typesetter was employed to reset, print, proof, and throw away the same copy. The union leader who negotiated this requirement is reported to have been a Mr. Bogus, and this redundant make-work typesetting was called ”bogus” type and added a word to the language. (There are other explanations for the word, but none contradicts this one). Even as late as the 1980s, most type was set on lead casting machines, and the production manager at the San Jose News complained that his reporters' stories were being retyped by ”400-dollar a month secretaries who type 80 words a minute and don't make mistakes, and then retyped at 40 words a minute on Linotype machines by 800-dollar a month Typesetters who do make mistakes.”

Input sample 4. (From *Out of Their Minds*, by Shasha, Lazere. Springer-Verlag, New York, 1995, page 99.)

Throughout his life, Knuth had been intrigued by the mechanics of printing and graphics. As a boy at Wisconsin summer camp in the 1940s, he wrote a guide to plants and illustrated the flowers with a stylus on the blue ditto paper that was commonly used in printing at that time. In college, he recalls admiring the typeface used in his math texbooks. But he was content to leave the mechanics of designing and setting type to the experts. ”I never thought I would have any control over printing. Printing was done by typographers, hot lead, scary stuff. Then in 1977, I learned about new printing machines that print characters made out of zeros and ones, just bits, no lead. Suddenly, printing was a computer science problem. I couldn't resist the challenge of developing computer tools using the new technology with which to write my next books.” Knuth designed and implemented TeX, a computer language for digital typography. He explored the field of typography with characteristic thoroughness. For example, he wrote a paper called ”The letter S” in which he dissects the mathematical shape of that letter through the ages, and explains his several day effort to find the equation that yields the most pleasing outline.

Here is what Sample 1 should look like when typeset with $M = 50$. Feel free to use this output to debug your code.

```
Total cost=5304
[43]    The first practical mechanized type casting
[38]    machine was invented in 1884 by Ottmar
[42]    Mergenthaler. His invention was called the
[43]    "Linotype". It produced solid lines of text
[46]    cast from rows of matrices. Each matrice was a
[45]    block of metal -- usually brass -- into which
[46]    an impression of a letter had been engraved or
```

```
[46]    stamped. The line-composing operation was done
[47]    by means of a keyboard similar to a typewriter.
[43]    A later development in line composition was
[46]    the "Teletypewriter". It was invented in 1913.
[42]    This machine could be attached directly to
[41]    a Linotype or similar machines to control
[46]    composition by means of a perforated tape. The
[45]    tape was punched on a separate keyboard unit.
[46]    A tape-reader translated the punched code into
[48]    electrical signals that could be sent by wire to
[50]    tape-punching units in many cities simultaneously.
[45]    The first major news event to make use of the
[31]    Teletypewriter was World War I.
```