1. *Kleinberg & Tardos* Chapter 5, question 1

2. Consider the problem of word-wrapping a paragraph of text. A paragraph is an ordered list of $n$ words, where word $w_i$ is $\ell_i$ letters long. You want to divide the paragraph into a sequence of lines, each containing at most $L$ letters. (No word is more than $L$ letters long.)

Suppose a line contains words $w_i \ldots w_j$. The total length $W(i,j)$ of this line is defined by

$$W(i,j) = j - i + \sum_{k=i}^{j} \ell_k.$$  

This length accounts for a single space between successive pair of words on the line. The *slop* $S(i,j)$ of this line is defined to be $L - W(i,j)$, the total number of unused spaces at the end of the line. Note that in any feasible solution, the slop of each line must be non-negative.

Just to make things concrete, consider the example paragraph “Now is the time for all good men.”, and suppose $L = 10$. One feasible solution is

```
Now is the
time for
all good
men.
```

This solution has four lines of lengths 10, 8, 8, and 4; the corresponding slops are 0, 2, 2, and 6.

Your goal is to find a division of the input paragraph into lines that minimizes the sum, over all lines except the last, of the *squared* slop of each line. (We omit the last line because it can in general be much shorter than the others.) For example, the total cost of the above solution is $0^2 + 2^2 + 2^2 = 8$.

Give a polynomial-time algorithm for this problem.

3. You are given a set of $n$ jobs, each of which runs in unit time. Job $i$ has an integer-valued deadline time $d_i \geq 0$ and a real-valued penalty $p_i \geq 0$. Jobs may be scheduled to start at any non-negative integer time (0, 1, 2, etc), and only one job may run at a time. If job $i$ completes at or before time $d_i$, then it incurs no penalty; otherwise, it incurs penalty $p_i$. Give a polynomial-time algorithm to schedule *all* jobs so as to minimize the total penalty incurred.