Lindsey Bieda and Joe Frambach
Greedy Algorithm Problems
9.11.2011

9. The input to this problem consists of an ordered list of n words. The length of the ith word is w_i, that is the ith word takes up w_i spaces. (For simplicity assume that there are no spaces between words.) The goal is to break this ordered list of words into lines, this is called a layout. Note that you can not reorder the words. The length of a line is the sum of the lengths of the words on that line. The ideal line length is L. No line may be longer than L, although it may be shorter. The penalty for having a line of length K is $L - K$. *The total penalty is the* ***sum*** *of the line penalties.* The problem is to find a layout that minimizes the total penalty.
Prove of disprove that the following greedy algorithm correctly solves this problem.

```
For i= 1 to n
     Place the ith word on the current line if it fits
      else place the ith word on a new line
```

Prove that the greedy algorithm is correct.
Proof: Assume $\exists$ input I $\ni$ GRE(I) < OPT(I), let OPT be the optimal solution that agrees with greedy for the most steps, k.

$$GRE(I) = \left.\begin{matrix} w_0 & w_1 & \dots & w_{k-1} & w_k \\ w_{k+1} & w_{k+2} & \dots & w_n & \end{matrix}\right\}, k = NL - \sum w, \text{ where N is the number of lines}$$

$$OPT(I) = \left.\begin{matrix} w_0 & w_1 & \dots & w_{k-1} \\ w_k & w_{k+1} & \dots & w_{n-1} \\ w_n & & & \end{matrix}\right\}, k = NL - \sum w$$

Let the first point of disagreement be at point k. We can construct OPT' as the following.

$$OPT'(I) = \left.\begin{matrix} w_0 & w_1 & \dots & w_k \\ w_{k+1} & w_{k+2} & \dots & w_{n-1} \\ w_n & & & \end{matrix}\right\}, k = NL - \sum w$$

At this point k is non-increasing, so $OPT' \geq OPT$, and we can construct OPT'' which will agree with greedy for an additional step.

$$OPT''(I) = \left.\begin{matrix} w_0 & w_1 & \dots & w_k \\ w_{k+1} & w_{k+2} & \dots & w_n \end{matrix}\right\}, k = (N-1)L - \sum w$$

$OPT \leq OPT' \leq OPT'' \leq \ldots = GRE\bot$

10. The input to this problem consists of an ordered list of n words. The length of the ith word is w_i , that is the ith word takes up w_i spaces. (For simplicity assume that there are no spaces between words.) The goal is to break this ordered list of words into lines, this is called a layout. Note that you can not reorder the words. The length of a line is the sum of the lengths of the words on that line. The ideal line length is L. No line may be longer than L, although it may be shorter. The penalty for having a line of length K is $L - K$. The total penalty is the maximum of the line penalties. The problem is to find a layout that minimizes the total penalty.
Prove of disprove that the following greedy algorithm correctly solves this problem.

```
For i= 1 to n
     Place the ith word on the current line if it fits
       else place the ith word on a new line
```

Counter-example:
Consider the following input: $I = [2, 1, 1]$ and $L = 3$,

	2	1	$k = 0$
GRE(I) =	1		$k = 2$
			$max(k) = 2$

	2		$k = 1$
OPT(I) =	1	1	$k = 1$
			$max(k) = 1$

13. Consider the following problem.
INPUT: Positive integers $r_1, \ldots, r_n$ and $c_1, \ldots, c_n$.
OUTPUT: An n by n matrix A with 0/1 entries such that for all i the sum of the ith row in A is r_i and the sum of the ith column in A is c_i, if such a matrix exists.
Think of the problem this way. You want to put pawns on an n by n chessboard so that the ith row has r_i pawns and the ith column has c_i pawns.
Consider the following greedy algorithm that constructs A row by row. Assume that the first $i-1$ rows have been constructed. Let a j be the number of 1s in the jth column in the first $i-1$ rows. Now the r_i columns with a maximum $c_j - a_j$ are assigned 1s in row i, and the rest of the columns are assigned 0s. That is, the columns that still needs the most 1s are given 1s. Formally prove that this algorithm is correct.

Assume $\sum r_1, \ldots, r_n = \sum c_1, \ldots, c_n$ otherwise it is *NOT* correct.

Prove that the greedy algorithm is correct.
Proof: Assume $\exists$ input I $\ni$ GRE(I) < OPT(I), let OPT be the optimal solution that agrees with greedy for the most steps, k.
k, in this case, represents a position in the matrix I, (k_x, k_y).

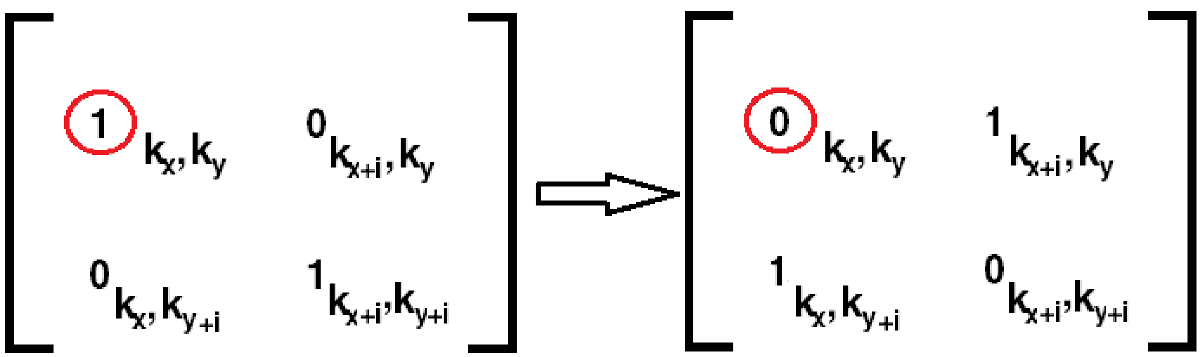

Figure 1: There must exist corresponding nodes in this configuration.

There must exist corresponding nodes in the configuration shown above (Fig.1). We can show that this must be the case.

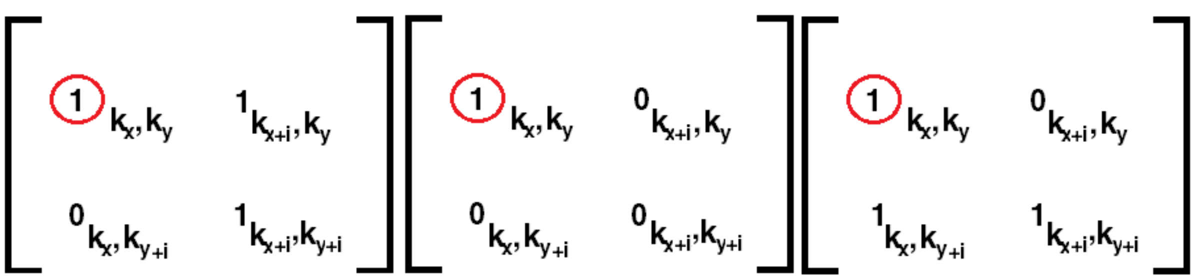

Figure 2: In each of these cases, swapping the values results in an incorrect solution

There are three cases to consider if the configuration is not found. In each of these cases (Fig. 2), swapping the values results in an incorrect solution. So the configuration (Fig.1) must exist.

OPT' is constructed by flipping the values in each node $[(k_x, k_y), (k_x + i, k_y), (k_x, k_y + i), (k_x + i, k_y + i)]$, as demonstrated in Fig. 1. OPT' results in an optimal solution since the sums of the rows and columns remain correct. The same holds true if $(k_x, k_y) = (k_x + i, k_y + i) = 0$ and $(k_x, k_y + i) = (k_x + i, k_y) = 1$.
We have now constructed an OPT' which agrees with OPT for one more step and is still optimal.

$OPT \leq OPT' \leq OPT'' \leq \ldots = GRE\perp$