CS 1510
Algorithm Design

Greedy Algorithms

Problems 6 and 7

Due September 3, 2014

Buck Young and Rob Brown

Problem 6

(a)
Input: An ordered sequence of words $S = \{w_1, w_2, ..., w_n\}$ where the length of the i^{th} word is w_i.
Output: Total penalty P as described by the problem statement (ΣP_i for each $P_i = K_i - L$) such that P is minimized.
Theorem: The given greedy algorithm G is correct.

Proof: Assume to reach a contradiction that there exists some input I on which G produces an unacceptable output. Let OPT be the optimal solution that agrees with G for the most number of steps. Let the first point of disagreement be the word w_{L_k} on line k. The only way OPT can disagree with G is if OPT moves w_{L_k} to a new line and G does not (obviously G will not fail to place a word on a line that it is able to).

Let OPT' be a modified version of OPT, such that the first word on each new line $(w_{L_k}, w_{L_{k+1}}, ..., w_{L_n})$ is moved to the previous line if possible. The logic governing this movement from the i^{th} line to the $(i-1)^{th}$ line is as follows.

for $i \geq k$ **do**
    if $w_{L_i} \leq (L - K'_{i-1})$ **then**
        if w_{L_i} is the ONLY word on the LAST line **then**
            remove the final line, as the word fits on the previous line
            P'_i changed from $L - w_i$ to 0
            $P'_{i-1} -= w_i$
            $\Delta P \leq 0$
        else
            Move w_i from L_i to L_{i-1}
            $P_i += w_i$
            $P_{i-1} -= w_i$
            $\Delta P = 0$
        end if
    else
        Cannot move w_{L_i} up
        $\Delta P = 0$
    end if
end for

Thus, any state that OPT' can reach will result in $\Delta P = P' - P \leq 0$. This tells us that $OPT' \leq OPT$, and we have that OPT' is correct. Additionally, we know that OPT' can fit w_k on the prior line, since G made this selection. So for at least one word we are able to make OPT' more like G. Now OPT' is both correct and agrees with G for one step more than OPT.

$\therefore$ OPT' contradicts the definition of OPT, and G is correct.

(b)

Problem 7

Algorithm: If a page is not in fast memory and an eviction must occur, evict the page that does not occur again or whose next use will occur farthest in the future.

Input: A sequence I = $\{P_1, P_2, ..., P_n\}$ of page requests

Output: Minimum cardinality of a sequence of evicted pages E

Theorem: The "farthest in the future" algorithm F is correct.

Proof: Assume to reach a contraction that there exists input I on which F produces unacceptable output.

Let F(I)=$\{P_{\alpha(1)}$, $P_{\alpha(2)}$, ..., $P_{\alpha(n)}\}$ be the output of F on input I
& OPT(I)=$\{P_{\beta(1)}$, $P_{\beta(2)}$, ..., $P_{\beta(n)}\}$ be the optimal solution that agrees with F(I) the most number of steps

Consider the first point of disagreement k between OPT and F

Let OPT' equal OPT with $P_{\alpha(k)}$ chosen at k instead of $P_{\beta(k)}$

Clearly OPT' is more similar to F (for one additional step)

To show that OPT' is still correct, consider the four possible cases for input I:

Case 1

In this case, neither $P_{\alpha(k)}$ nor $P_{\beta(k)}$ exist in the input after time-step k. Therefore $|E|$ and $|E'|$ do not change.

Case 2

In this case, only $P_{\beta(k)}$ exists in the input after time-step k. Since $P_{\beta(k)}$ is needed at some point in the future (and OPT has evicted it at time-step k), OPT is guaranteed to need at least one additional eviction. Therefore $|E| > |E'|$.

Case 3

In this case, both $P_{\alpha(k)}$ and $P_{\beta(k)}$ exist in the input after time-step k. Therefore $|E|$ must account for an eviction to bring $P_{\beta(k)}$ back into fast memory and $|E'|$ must account for an eviction to bring $P_{\alpha(k)}$ back into fast memory.

Case 4

In this case, only $P_{\alpha(k)}$ exists in the input after time-step k. By the definition of F, if $P_{\beta(k)}$ did not exist in the input after time-step k then F would have chosen it. Therefore this is an impossible state.

In all cases, $|E| \geq |E'|$. Therefore (since we are minimizing) OPT $\leq$ OPT'

$\therefore$ We have reached a contradiction because OPT' agrees with F for one additional step (and still produces a correct output) despite OPT being defined as agreeing with F for the most number of steps.