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 algoirthm G correctly solves this problem.

Proof: Assume to reach a contradiction that there exists an input on which G produces an unacceptable output.

Problem 5

(a)
Input: Locations x_1, x_2, ..., x_n of gas stations along a highway.

Output: Stoppage time at each station which minimizes the total time which you are stopped for gas.

Theorem: The given "next-station" algorithm (NS) is correct / optimal.

Proof: Assume to reach a contradiction that there exists an input on which NS produces an unacceptable output.

Let S_i be the stoppage time at each station
& NS(I)=$\{S_{\alpha(1)}$, $S_{\alpha(2)}$, ..., $S_{\alpha(n)}\}$ be the output of NS on input I
& OPT(I)=$\{S_{\beta(1)}$, $S_{\beta(2)}$, ..., $S_{\beta(n)}\}$ be the optimal solution that agrees with NS(I) the most number of steps

Let x_k be the first stop of the trip upon which NS(I) differs from OPT(I)

By the definition of NS, filling up for any time less than $S_\alpha(k)$ would not get you to the next station and since $S_{\alpha(k)} \neq S_{\beta(k)}$, then we know that $S_{\beta(k)} > S_{\alpha(k)}$

Let time t be defined as the difference between those stoppage times, t = $S_{\beta(k)}$ - $S_{\alpha(k)}$

Therefore $S_{\beta(k)} = S_{\alpha(k)}$ + t:

Let OPT'(I) = OPT(I) with time t shifted to the next stop:

$$S'_k = S_k \text{ - t}$$
$$S'_{k+1} = S_{k+1} + t$$

Clearly, we haven't changed the total stoppage time and we know that we can reach the next stop by the definition of NS. Furthermore we add the difference to the stop at x_{k+1}, so we will undoubtably be able to reach the station at x_{k+2} – and all future stations.

∴ We have reached a contradiction because OPT' agrees with NS for one additional step despite OPT being defined as agreeing with NS for the most number of steps.

(b)
Input: Locations x_1, ..., x_n of gas stations along a highway.

Output: Stoppage time at each station which minimizes the total time which you are stopped for gas.

Theorem: The given ”fill-up” algorithm (FU) is not correct or optimal.

Proof:

Consider the input I = $\{x_1, x_2\}$ where $x_2 - x_1 = \frac{1}{2}\frac{C}{F}$

Note that to reach x_2 from x_1 it would take $F(x_2 - x_1) = F(\frac{1}{2}\frac{C}{F}) = \frac{1}{2}C$ liters of gas

At x_1 the tank is filled to capacity according to FU, taking $\frac{C}{r}$ minutes. But it is possible and correct to reach x_2 using only $\frac{1}{2}C$ liters, making the optimal time $\frac{1}{2}\frac{C}{r}$ minutes.

$\therefore$ FU is not correct or optimal.

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.