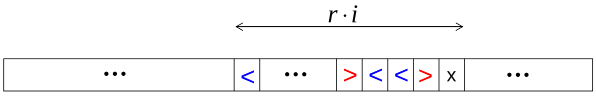
6.851 Advanced Data Structures (Spring'07)

Prof. Erik Demaine TA: Oren Weimann

Problem 1 – Solution

Move-To-Front Variations. Assume that there are n elements and as we start, the last k items in our list are $k, k-1, \ldots, 1$. Consider the sequence of requests: 1,2,3,...,k,1,2,3,...,k,1,2,3,...k,... If we move the accessed item forward by k locations then the time to serve this request sequence of length m is $\Theta(nm)$. However, the Move-To-Front heuristic serves this request sequence in $\Theta(km)$ time. The competitive ratio is therefore $\frac{n}{k}$, which is unbounded as k is a constant.

We next show that moving x forward by $r \cdot i$ locations gives a competitive ratio of $2/r$. We set the potential function to be $\Phi = \frac{1}{r} \cdot \#$ of inversions with OPT, and we label each of the $r \cdot i$ elements in front of x with $<$'s and $>$'s.



When accessing x, $\Delta\Phi = \frac{\# <'s - \# >'s}{r}$ and the actual cost of the access is $i + 1 = \frac{ri}{r} + 1 = \frac{\# <'s + \# >'s}{r} + 1$. Therefore, the amortized cost is $\Delta\Phi$+actual cost $= \frac{2\# <'s}{r} + 1 \le \frac{2}{r} \cdot OPT - 1$.

Splay Trees Bounds. By the working-set theorem, if x is requested in times $i_1, i_2, \ldots, i_{f_x}$ then the total cost of all accesses to x is

$$O\Big(\sum_{i \in \{i_1, \ldots, i_{f_x}\}} 1 + \lg t_i(x)\Big) = O\Big(f_x + \sum_{i \in \{i_1, \ldots, i_{f_x}\}} \lg t_i(x)\Big).$$

This last sum is maximized when all the $t_i(x)$'s are equal and we know that $\sum_{i \in \{i_1, \ldots, i_{f_x}\}} t_i(x) \le m$ so

$$\sum_{i \in \{i_1, \ldots, i_{f_x}\}} \lg t_i(x) \le \sum_{i \in \{i_1, \ldots, i_{f_x}\}} \lg\Big(\frac{m}{f_x}\Big) = f_x \lg \frac{m}{f_x}$$

Therefore, the amortized cost of accessing an element is

$$O\Big(\frac{1}{m} \sum_x \Big(f_x + f_x \lg \frac{m}{f_x}\Big)\Big) = O\Big(1 + \frac{1}{m} \sum_x f_x \lg \frac{m}{f_x}\Big).$$