# 6.851 Advanced Data Structures (Spring'12)

Prof. Erik Demaine TAs: Tom Morgan, Justin Zhang

## Problem 3 *Sample solution*

**Geometric basics.** To show that there is a binary search tree supporting any $m$ searches on $n$ items in $O(m \log n)$ total time, we will show that given a set of $m$ points with $y$-coordinates $1..m$ and $x$-coordinates in the set $\{1..n\}$, we can insert at most $O(m \log n)$ additional points such that the total set is arborally satisfied. By making the set arborally satisfied, it will represent a valid BST execution, and by having a total of $O(m \log n)$ points in the set, it means that the tree touches $O(m \log n)$ points in performing its $m$ queries, thus it runs in $O(m \log n)$ total time.

Our procedure for inserting these points is as follows. Initially, we observe that given a set of points with $x$-coordinates in the range $[1, a]$, if for each point $(x, y)$ we insert a new point $(a/2, y)$ then any pair of points $(x_i, y_i)$, $(x_j, y_j)$ such that $x_i < a/2 < x_j$ will be arborally satisfied as the rectangle they generate will cross two of the points. We can now recursively perform the same procedure on the range $[1, a/2)$ and $(a/2, a]$ until we yield a region with all points having the same $x$ value. In doing so, it is clear we will have created an arborally satisfied set. To show that we have created $O(m \log n)$ points, we observe that for each of our $m$ original points, we create $\log n$ new points. This is because each time we recurse on the two sides $[1, a/2)$ and $(a/2, a]$, this original point will contribute to only one side, so the number of points created in this row is equal to the depth of the recursion, and as we are only working with integers in the range $1..n$, the depth is $\log n$.

**Working-set is harder.** Under the working-set property, if $x$ is requested in times $i_1, i_2, \ldots, i_{f_x}$ then the total cost of all accesses to $x$ is

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

We know that $\sum_{i \in \{i_1, \ldots, i_{f_x}\}} t_i(x) \leq m$, so this last sum is maximized when all the $t_i(x)$'s are equal to $m/f_x$ (by AM-GM or Jensen's inequality), thus

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

Therefore, the amortized cost of accessing an element is

$$O\left( \frac{1}{m} \sum_x f_x \lg \frac{m}{f_x} \right) = O\left( \sum_x \frac{f_x}{m} \lg \frac{m}{f_x} \right).$$

Thus, since $p_x = \frac{f_x}{m}$, the entropy bound holds.