Przemysław Uznański
Chih-Hung Liu

March 21, 2017

Advanced Data Structures

Spring Semester 2017

Exercise Set 5

Exercise 1:
Present an implementation of your favorite data structure in your favorite purely functional programming language (for example, *balanced search trees* in *Haskell*). Discuss its full persistency.

Exercise 2:
(Dominance Query) For two points, $p = (x_p, y_p)$ and $q = (x_q, y_q)$, p is said to *dominate* q if $x_q \leq x_p$ and $y_q \leq y_p$. Consider a set S of n points in the plane, and process S, in $\mathcal{O}(n \log n)$ time, such that for a query point $p = (x_p, y_p)$, the points in S that are dominated by p can be answered in $\mathcal{O}(\log n + k)$ time, where k is the output size.

Hint:

- For each point $q \in S$, project an upward ray from q.
- Project a leftward ray l from the query point p and find all the upward rays intersected by l, from which the points dominated by p can be obtained.
- Move a vertical line from left to right, and store the vertical order.

Exercise 3:
Consider a set S of n disjoint axis-parallel rectangles and process S, in $\mathcal{O}(n \log n)$ time, such that for a query vertical line segment l, the rectangles in S intersected by l can be answered in $\mathcal{O}(\log n + k)$ time, where k is the output size.
Hint: Use the segment tree and the plane sweep paradigm.

Exercise 4:
(Potential Analysis of Full Persistence) Show that the usage 2(d+p+1) mods per node allows a constant amortized update time, where d and p are the out-degree and the in-degree of a node, respectively.
Hint: potential $\Phi = c \cdot \sum_{\text{node}} \big((d+p+1) - \min\{d+p+1, \# \text{ empty mods slots}\}\big)$, where c is a suitable constant.