6.851 Advanced Data Structures (Spring'12)

Prof. Erik Demaine

Problem 10 *Due: Thursday, May 3*

Be sure to read the instructions on the assignments section of the class web page. Remember to keep your solutions to one page!

Compact balanced parentheses. Recall that in lecture we introduced the balanced-parentheses representation of a binary trie. A common query when using this representation is `match`(i), which returns the index of the close parenthesis matching the open parenthesis at index i. In this problem we will develop a compact data structure to answer these queries (though a succinct data structure is known).

1. Let S be a string of balanced parenthesis. Divide S into blocks of size B. We will call an open parenthesis *far* if its matching parenthesis is in a different block. A far parenthesis is a *pioneer* if its matching parenthesis is in a different block than the matching parenthesis of the previous far parenthesis in S. Prove that there are $O(|S|/B)$ pioneer parenthesis.
2. Develop and analyze a static data structure that, given a string of balanced parenthesis S, answers `match` queries in $O(1)$ time and uses $O(|S|)$ bits of space.

6.851 Advanced Data Structures
Spring 2012