Algorithms in Bioinformatics II, SS2004
Assignment sheet # 6

Daniel Huson

May 23, 2004

1 Size of the T data-array (2 points)

What is the maximum length required for the data-array T in the WOTD algorithm, as a function of n, the length of the text?

2 Memory requirement (1 point)

Why does the algorithm that builds the WOTD suffix tree require *two* copies of the auxiliary array $suffixes$?

3 Worst-case runtime complexity of WOTD algorithm (2 points)

What is the worst-case runtime complexity for building the complete WOTD suffix tree, and why?

4 Finding all occurrences of a query in a WOTD suffix tree (5 points)

Please download `www-ab.informatik.uni-tuebingen.de/teaching/ss04/abi2/java/assign06.zip`. The file `albi2.suffixtree.WOTDBase.java` contains code for building a WOTD suffix tree and the file `albi2.suffixtree.RunWOTD.java` contains code for running a text-query program. Your task is to implement a method `findWOTD` in the class `albi2.suffixtree.WOTD.java` that finds and reports all occurrences of a given query string.

Assignments due: **Tuesday, June 1, 10am**