

Algorithms in Bioinformatics II, SS2003

Assignment sheet # 7

Daniel Huson

June 11, 2003

In this assignment sheet, the goal is to write the algorithm that finds occurrences of a query in a text using a suffix tree built by the WOTD algorithm. Please download

`www-ab.informatik.uni-tuebingen.de/teaching/ss03/abi2/java/assign07.zip`

and modify the file `WOTD.java`. Modification of `WOTDBase.java` should not be necessary.

1 Implementation of the find method (4 points)

In `WOTD.java`, implement a method `find` that returns all occurrences of a query in the given text.

2 Lazy implementation (3 points)

Modify the class `WOTD.java` so that the tree is constructed lazily. That is, first add a boolean parameter to the constructor for `WOTD` that determines whether the tree is to be built lazily or not and then modify the constructor and your `find` method accordingly.

3 Reading the text and queries (3 points)

Modify the program so that a text is read from an input file named on the command line and then queries are read from standard input. Query results should be written to standard output.

Run the algorithm on the text `mississippiz` (using `z` as the terminator symbol) and a number of queries that show that your algorithm works correctly.

Discuss an example for which the lazy implementation performs much better than the non-lazy variant.

Note As I just wrote this implementation of the WOTD algorithm last night, it may still have minor bugs. Please fix any bugs found!

Assignments due: **Monday, June 15, 10am**