Homework 12

Exercises 1-2

Two reviews are done. My student number is C28533.

Exercise 5

Let's perform the Burrows-Wheeler Transform on string **CGGTCGCT$** step by step.

First, we should build a table of all circular shifts of the string.

The Shifts Table

C	G	G	T	C	G	C	T	$
$	C	G	G	T	C	G	C	T
T	$	C	G	G	T	C	G	C
C	T	$	C	G	G	T	C	G
G	C	T	$	C	G	G	T	C
C	G	C	T	$	C	G	G	T
T	C	G	C	T	$	C	G	G
G	T	C	G	C	T	$	C	G
G	G	T	C	G	C	T	$	C

Then we should sort the rows by the first column.

The Sorted Shifts

$	C	G	G	T	C	G	C	T
C	G	C	T	$	C	G	G	T
C	G	G	T	C	G	C	T	$
C	T	$	C	G	G	T	C	G
G	C	T	$	C	G	G	T	C
G	G	T	C	G	C	T	$	C
G	T	C	G	C	T	$	C	G
T	$	C	G	G	T	C	G	C
T	C	G	C	T	$	C	G	G

The last column of the table is our Burrows-Wheeler Transform of the original string: **TT$GCCGCG**.

To reconstruct the original string, we, firstly, should sort the characters in the string. Thus, we get the first and last columns of the original table.

The Transformed String Next To Itself Sorted

$	T
C	T
C	$
C	G
G	C
G	C
G	G
T	C
T	G

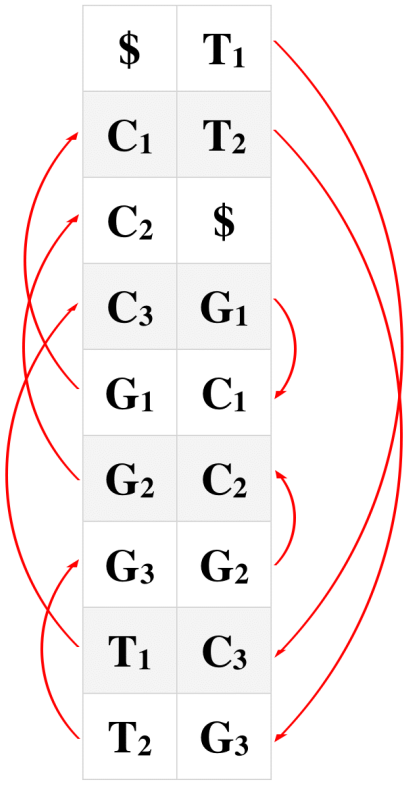
Now, we should enumerate the similar characters by the order of their appearance.

The Same Columns But With Their Characters Enumerated

$	T_1
C_1	T_2
C_2	$
C_3	G_1
G_1	C_1
G_2	C_2
G_3	G_2
T_1	C_3
T_2	G_3

Finally, making use of the circularity of the shifts, we can recreate the original string by following the pairs of the neighboring characters in the rows.

The Order Of Visiting The Neighboring Characters



Starting from $ and traversing the table until we encounter $ again (the order is $T_1C_3G_1C_1T_2G_3G_2C_2$), and then reversing the result, we get the original string: **CGGTCGCT$**.

Usage

The Burrows-Wheeler Transform is used for data compression, for example, in the Unix compression utility bzip2. This technique utilizes the property of the transform to create long sequences of similar characters and thereby replaces such sequences with a repeated character and the number of its occurrences. Another application of BWT is compression of genomic databases since a human genome has a lot of repeats (the only characters are C, T, G, A). Other tasks that make use of BWT include sequence alignment, sequence prediction and more.

Exercise 7

- **Which topics were most useful?**

For me, it was useful to refresh the skills of comparing the performance of different algorithms via diffrenet kinds of plots; comparing algorithms theoretically (via complexities); implementing lists, trees, graphs and different algorithms on them (it was useful to illustrate the algorithms by hand, which, among other things, gave me a good intuitive understanding of how balanced trees work); hashing methods; automatons; etc... It's easier to say what was boring and superfluous.

- **What needs to be covered better in the course?**

My prior knowledge in algorithmics was extremely helpful to me, because the lecture slides turned out to be very obscure and unintelligible, in many ways just useless. The lectures themselves did not follow the slides (the "stream of consciousness" style, which was not bad and provided a good intuitive understanding of the topics), but concentrated on very few things. Thus, lots of details remained uncovered, so my most useful learning material was GeeksForGeeks. Some topics were especially bad covered: for example, automatons and succinct data structures. Transforming regular expressions into NFA, and then into DFA, and then minimizing the DFA is such a complex process and is quite a topic in itself. So personally, I would split it into two lectures rather than trying to fit everything into 1.5 hours or 100 slides. Concerning succinct data structures, the slides were (frankly speaking) bad and there was almost impossible to find any materials on the topic, which seems very narrow and unpopular.

- **Are there some topics that would need more practical implementation assignments?**

Rather the other way round.

- **Are there some topics that got too much attention (e.g. too boring or otherwise already well known)?**

Succinct data structures.

- **Other remarks:**

One of the main problems of the homeworks (which generally were good with especially interesting bonus tasks) is the formulations of exercises. Some tasks took me more time to interpret them than to actually implement them. Some of them were too large, neighboring with the others that were too easy (for example, building an automaton from regex in Homework 11 costed 1 point, whereas easy bonus tasks costed 3 points each). But after all, the course was useful and the practice teacher was very intelligent, kind and understanding.