Tandem Repeat Detection

This project implements Main & Lorentz's (1985) algorithm for tandem repeat/square detection.

`make` generates the following executables:

Executable	Time Complexity	Summary
tr.o	O(nlogn)	Their main algorithm
tr_fast.o	O(n)	Speedup to above through preprocessing
tr_bf.o	O(n^2)	Naive approach for comparison

Input Format

Each executable reads data from stdin following this format:

```
num_tests start_char alphabet_size
query_string_for_test_1
query_string_for_test_2
...
```

and outputs 'YES' for each test case that contains a tandem repeat, and 'NO' otherwise. See `tests/in/100w_5n_4k.in` for a simple example.

The alphabet is defined as the alphabet_size ASCII characters beginning from the start_char. For example, start_char = 'a' and alphabet_size = 4 gives an alphabet of {'a', 'b', 'c', 'd'}.

Note the following input requirements, which are not validated but must be followed:

- All query strings must only use characters from the alphabet
- The smallest allowed start_char is "!"
- The alphabet size should be defined so that the largest character doesn't go beyond "~"

Scripts

All testing scripts can be run using:

```
bash test_scripts/test.sh
bash test_scripts/test_manual.sh
bash test_scripts/test_time_complexity.sh
```

Script	Description	Optional Argument
test.sh	Compares O(nlogn) algorithm to O(n^2) on randomly generated strings	`./tr_fast.o` to test O(n) alg
test_manual.sh	Runs O(nlogn) algorithm on manually created strings with known answers	`./tr_fast.o` to test O(n) alg
test_time_complexity.sh	Times O(nlogn) and O(n) algorithms on long generated squarefree strings	`generate` to recreate input strings