Chris Quenzer

Project 2 Writeup

Pseudo-code. Please provide the pseudo-code for your algorithm.

Edit Distance:

```
edit_dist(m,n){

	table[0][0] = 0; //base cases
	for (i = 1; i < m + 1; i++)
	{
		table[i][0] = delete_cost;
	}
	for (j = 1; j < n + 1; j++)
	{
		table[0][j] = delete_cost;
	}

	for (int i = 1; i < m + 1; i++)
	{
		for (j = 1; j < n + 1; j++)
		{
			table[i][j] = min(delete_cost, insert_cost, align_cost)
			min_cost = table[i][j];
			//keep track of backtrace table
			if (min_cost  == align_cost) //align --- DIAG
			{
				back[i][j] = DIAG;
			}
			else if (min_cost == delete_cost) //delete --- DOWN
			{
				back[i][j] = D;
			}
			else if (min_cost == insert_cost) //insert --- LEFT
			{
				back[i][j] = L;
			}
		}
	}
	return min_cost;
}
```

Backtrace:

```
align(m,n){

	i = m + 1;
	j = n + 1;

	while (i > 1 && j > 1)
	{
		if (back[i - 1][j - 1] == DIAG) // diag
		{
			str1_new = str1[j - 1] + str1_new;
			str2_new = str2[i - 1] + str2_new;
			i--;
			j--;

		}
		else if (back[i - 1][j - 1] == D) // down
		{
			str1_new = "-" + str1_new;
			str2_new = str2[i - 1] + str2_new;
			i--;
		}
		else if (back[i - 1][j - 1] == L) // left
		{
			str1_new = str1[j - 1] + str1_new;
			str2_new = "-" + str2_new;
			j--;
		}
	}

	if (j > 1)
	{
		Add remaining chars from str1 to str1_new;
		Add '-' to the rest of str2;
	}

	if (i > 1)
	{
		Add remaining chars from str2 to str2_new;
		Add '-' to the rest of str1;
	}
	return str1_new, str2_new;
}
```

Asymptotic Analysis of run time. Please analyze the runtime for your algorithm.

- **Edit Distance:**
  $T(m,n) = T(m \times n) + c \rightarrow T(m,n) = O(mn)$

  The algorithm must go through and calculate the optimal choice for each and every element in a table of m x n size. The step of filling in the element entry is constant time, so the final running time is O(mn).

- **Backtrace:**
  $T(m,n) = T(m + n) + c \rightarrow T(m,n) = T(m + n)$

The algorithm, at worst, must loop through and assign each letter m times, and then fill in the rest of the string n times. This is if i gets to be 0 before j decrements at all, thus causing an added time of n onto the already m time. This causes a running time of O(m + n).

Reporting and plotting the runtime. Describe how the running time is measured in your experiments. What type of machine is used for the experiments. Is there any part excluded from the runtime measurements (e.g., output of the alignment)? Plot the running times as a function of the input size. Label your plot (axes, title, etc.). Include an additional plot of the running times on a log-log axis [1]. Note that if the slope of a line in a log-log plot is m, then the line is of the form $O(x_m)$ on a linear plot. Determine the slope of the line in your log-log plot and from the slope, infer the experimental running time for your algorithm.

- **How the data is recorded:**
  Using the <ctime> library, the clock cycles from the beginning of the code block to the end are grabbed, then the difference is divided by clocks per second to get the time in milliseconds.

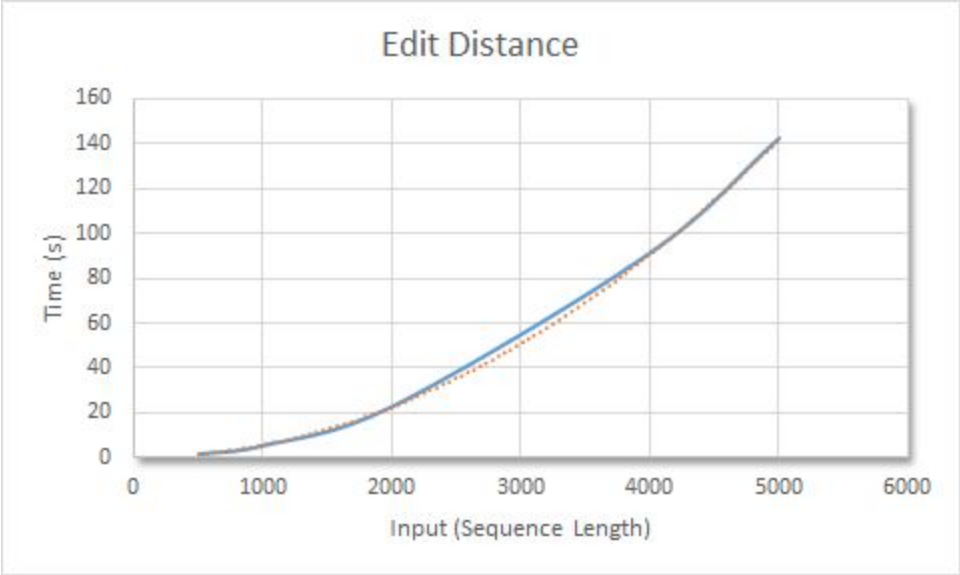
```
#include <ctime>
….

ofstream outfile_t;
outfile_t.open("timing.txt", ios::out | ios::app);
outfile_t << endl << "--- Align ---" << endl;
start_s = clock();

// Code to time

stop_s = clock();
outfile_t << "time: " << (stop_s - start_s) / double(CLOCKS_PER_SEC) * 1000 << endl;
outfile_t.close();
```

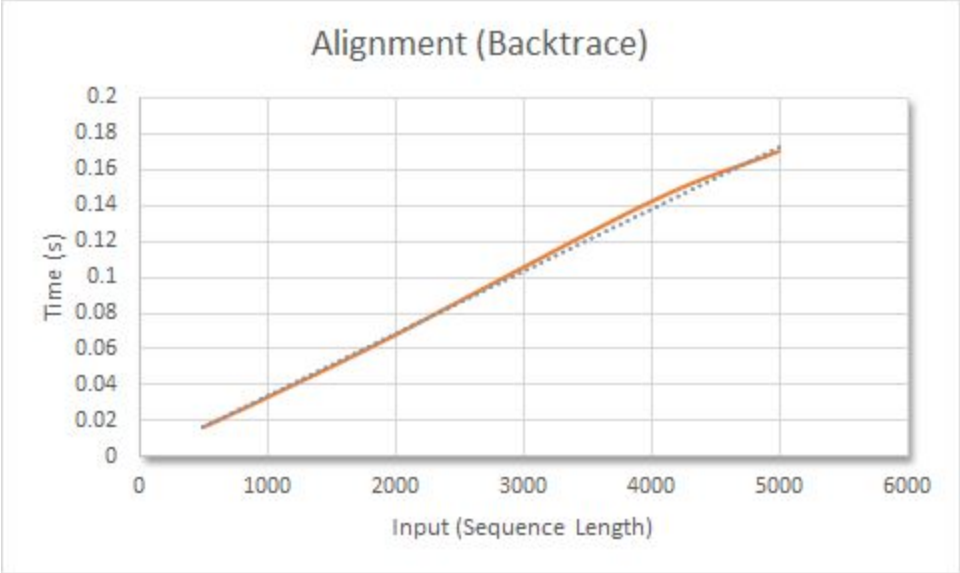
- **Machine Used:** Desktop tower
  - Intel Core i5-3570K @ 3.4 GHz
  - 16 GB RAM
  - Nvidia GTX 970 GPU
  - 64-bit operating system (Windows)

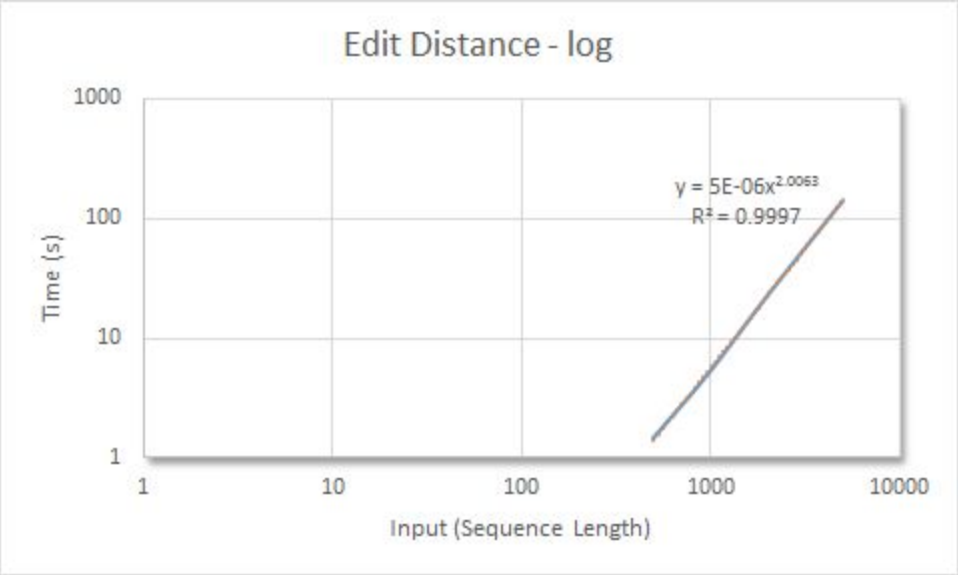
The only parts excluded from the timing are the writing of the aligned strings to the output file. The timed portions are the ones doing the actual algorithm.

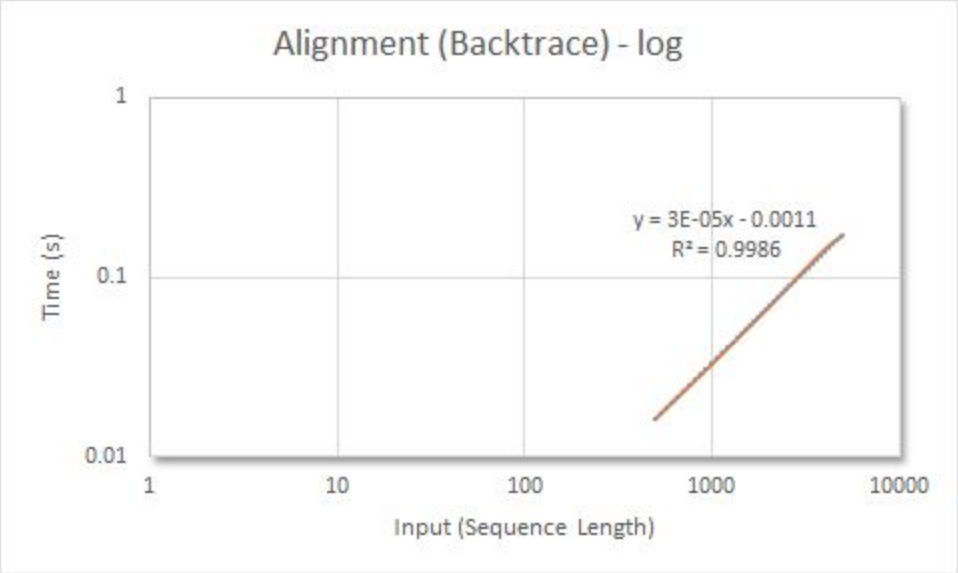
Data:

Sequence Length (n)	Edit Distance: Avg Time (s)	Backtrace: Avg Time (s)
500	1.446	0.016
1000	5.299	0.033
2000	22.788	0.068
4000	91.232	0.142
5000	142.330	0.170


Edit Distance
160
140
120
100
80
60
40
20
0
Time (s)
0
1000
2000
3000
4000
5000
6000
Input (Sequence Length)


Alignment (Backtrace)
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Time (s)
0
1000
2000
3000
4000
5000
6000
Input (Sequence Length)





Edit Distance Log Slope	Backtrace Log Slope
1.873	1.009

Experimental Running Time: Edit Distance - O(1.873m), Backtrace - O(1.003m)

Interpretation and discussion. Discuss the runtime plot. Do the growth curve match your expectation based on their theoretical bounds?

The runtime plots match almost exactly to what they theoretically should be. Using best fit lines on each of the data sets, a power fit to the Edit Distance algorithm produced a plot with x^2 slope. The R value is 0.9997, so a very close fit. The Backtrace data produced a plot fit with a constant slope, and R value of 0.9986. The slope calculated for the

Edit Distance algorithm was 1.873, so if the experimental run time is determined by O(x^m), the run time of this experiment would be O(1.873^m), or nearly O(2^m). The slope calculated for the Backtrace part was 1.003, so if the experimental run time is determined by O(x^m), the run time of this experiment would be O(1.003^m).