Mitchell Van Braeckel (mvanbrae@uoguelph.ca) 1002297
Mar 11th, 2019
CIS*3490: Analysis and Design of Computer Algorithms – A3

2.4 Analyze the performance of your brute force and Horspool Algorithms. (10%)
The performance parameters are the number of pattern shifts and running time. To compare two programs, choose ten search patterns of different lengths, and search them by using the programs separately. For each pattern, calculate the ratios of the performance parameters of the two programs. Then, for all the ten patterns, calculate average ratios, and compare and analyze the performance of the two programs in terms of the ratios. Very briefly write your comparison and analysis in the readme file submitted with your programs.

```
* Assumption: since the A3 handout description just says to choose 10
search patterns of different lengths, I will assume that it doesn't matter
how many times the search pattern occurs.
```

Exhaustive Brute Force Performance Analysis

Word	Occurrences Found	Pattern Shifts	Execution Time (seconds)	Ratio (shifts * time)
WWII	1	3300588	0.041	135324.108
RSPCA	1	3300587	0.049	161728.763
tongue	1	3300586	0.051	168329.886
planninga	1	3300583	0.054	178231.482
anumerical	1	3300582	0.059	194734.338
pluralistic	1	3300581	0.056	184832.536
lifelonglove	1	3300580	0.057	188133.060
scholarshipmay	1	3300578	0.054	178231.212
CanadaDepartment	1	3300576	0.048	158427.648
facultysupervising	1	3300574	0.054	178230.996
AVERAGE:	**1**	**3300582**	**0.052**	**172620.403**

NOTE: Averages calculated using column (rounded)

Horspool Performance Analysis

Word	Occurrences Found	Pattern Shifts	Execution Time (seconds)	Ratio (shifts * time)
WWII	1	826644	0.108	89277.552
RSPCA	1	665487	0.097	64552.239
tongue	1	604422	0.092	55606.824
planninga	1	418830	0.078	32668.740
anumerical	1	424284	0.082	34791.288
pluralistic	1	378136	0.073	27603.928
lifelonglove	1	339856	0.073	24809.488
scholarshipmay	1	310382	0.072	22347.504
CanadaDepartment	1	295042	0.070	20652.940
facultysupervising	1	264662	0.067	17732.354
AVERAGE:	**1**	**452775**	**0.081**	**39004.286**

NOTE: Averages calculated using column (rounded)

The ratio (multiplication) calculation allows a visual statistic for the trend of efficiency since we want both a low execution time and number of pattern shifts.

For exhaustive brute force, the number of pattern shifts is always *n*-*m* (where *n* is the length of the string, and *m* is the length of the pattern), so the execution time won't very greatly either and will remain about the same for most patterns. Whereas for Horspool's Algorithm, even though the execution time is longer (especially for smaller patterns), it becomes much faster as the pattern size increases. Also, it always performs drastically less pattern shifts, and this number continues to decrease as the pattern size increases. Furthermore, since both the number of pattern shifts and execution continue to decrease as the patter size increases, in turn the ratio also decreases. These trends are shown well by our average ratios because Horspool's value is approximately 4.5 times smaller than brute force's. With perfect execution of Horspool's, it essentially divides the text by the size of the pattern in the best-case scenario where efficiency basically becomes (text length) / (pattern length).

Therefore, Horspool's Algorithm is obviously much more efficient, especially when searching larger strings with larger patterns since the number of pattern shifts is drastically reduced.