Name: Maxwell Aladago: ID:15992018

1 Task1: Data Pre-Preprocessing

I performed two pre-processing steps for this task: splitting each record into a list of values and separating the records into the two categories, *cumulative_cases* and *cumulative_deaths*. Only dates and the values were extracted from the records. Both steps were executed concurrently while the file was opened for reading.

1.1 Difficulty Level 1

I answered (a) and (b) by finding the index of the first occurrence of the maximum cumulative value of cases or deaths respectively. I then returned the value at that index in the dates list for that category. I answered (c) by adding 42 days to the answer in (a), the date a case last occurred. For (d) and (e), I iteratively tracked the highest rate observed over time, the date of the highest rate as well as the rates(for (f) and (g)). I updated these values using the rule

- if $(r > P_r) : P_r := r, P_rDate := curDate$ Where $r = (ccv - prevCv)/(curDate - prevDate)$ and P_r is the highest rate and P_rDate is the date for this highest value. P_rDate is returned as answer to either (d) or (e) depending on the class.

1.2 Difficulty Level 2

In answering (d) and (e) above, I saved the rates. Local peaks were then extracted from these values using the rule: if $f(x) < f(y) > f(z), \forall_{x,y,z} \quad x < y < z$, where x, y, z are dates, $f(y)$ is considered a local peak. All y_{is} are returned at the end as the list of peak dates. Their count is taken as the answer to (f) or (g) depending on the category.

2 Task2: Data pre-processing

I created a list called *pattern* from the partial data file and a dictionary of the form $dic = \{local : \{indicator : [[dates], [values]]\}\}$ from the complex data file. *local* is a combination of Country and Locality. That grouped the records in two different levels: into localities and into indicators within each locality. Each *indicator* has a list of dates and a list of cumulative values.

2.1 Main Task

I solved *Task2* by using either Knuth-Morris-Pratt(KMP)(Mandumula, 2011) algorithm or Boyer-Moore(BM) algorithm(Plaxton, 2005) depending on the length of the partial data file. I used KMP for shorter patterns since BM does not perform so well for small patterns(Moore, 2018). This made the algorithm adaptable to the size of the partial data file. I used KMP when the size of the pattern is ¡ 10 and Boyer-Moore for all other patterns. I chose the 10 arbitrarily.

For KMP, I created a suffix table from the pattern as described in (Mandumula, 2011). The suffix tables specify the skip steps to take whenever a mismatch occurs. For BM, I created a bad item lookup table and a good suffix lookup table. These tables serve similar purposes as the KMP suffix table. The details of constructing these tables are specified in Plaxton(2005). It is good to indicate that, I do not construct the suffix table when using Boyer-Moore. In a similar manner, only the suffix table is constructed if KMP is the algorithm being used.

After preprocessing, I iteratively searched for the pattern in $values_{ij}$ Where $values_{ij} = dic[local_i][indicator_{ij}][1]$ iFF the the size of $values_{ij} \geq$ the size of the pattern. $i \in \{1, 2, ..., m\}$ and $j \in \{1, 2\}$. . I terminated the search on the first instance of success. If success, the *local, indicator and start_date*. Start_date is retrieved from the dictionary using the local, the indicator and the index returned by he search. The overall runtime when using KMP is $O(mn + (n + k))$. BM has a runtime of $O(m(n/k) + n)$ in typical cases. m is the number of distinct localities, n is the average number of entries per indicator and k is the size of the pattern such that $n \geq k$

For both tasks, I worked under the assumption that since it is a time series data, the data is pre-sorted according to dates.

References

Mandumula, K. K. (2011). Knuth-morris-pratt. *Indiana State University*. Retrieved from `http://cs.indstate.edu/~kmandumula/kranthi.pdf`

Moore, J. S. (2018). The boyer-moore fast string searching algorithm. *University of Texas at Austin*. Retrieved from `http://www.cs.utexas.edu/users/moore/best-ideas/string-searching/index.html`

Plaxton, G. (2005). String matching: Boyer-moore algorithm. *University of Texas at Austin*. Retrieved from `http://www.cs.utexas.edu/~plaxton/c/337/05f/slides/StringMatching-4.pdf`