Career Fair 2018 Programming Competition Report
Maxwell Aladago: ID:15992018

1 Task1: Data Pre-Preprocessing

I performed two pre-processing steps for this task. Splitting each record in the data into lists and separating the records into the two categories, *cumulative_cases* and *cumulative_deaths*. Both steps were executed concurrently whilst the file was opened for reading.

1.1 Difficulty Level 1

In order to answer the questions in the category, I maintained variables for the last date a positive a record was made, the cumulative value of the record, the rates over time as well as the peak rates and their dates. These variables are updated iteratively using the rules

1. if $(ccv > lastcv) : lastcv := ccv, lastcv\_date := ccv\_date$ where *ccv = current cumulative value, lastcv = last cumulative value.* ccv_date is returned as the answer to (a) or (b) depending on the class.
2. if $(r > p\_r) : p\_r := r, p\_r\_date := prev\_date - cur\_date$ Where rate, $r = (ccv - prev_c v)/(cur\_date - prev\_date)$ and p_r is the highest rate and p_r_date is the date for this peak. p_r_date is returned as answer to (d) or (e) depending on the class.

The rules above were implemented in one function and then applied to each of the classes independently. (c) was computed by adding 42 days to results of 1. above for *cumulative_cases*.

1.2 Difficulty Level 2

In answering (d) and (e) above, the rates are saved. Local peak rates are then extracted from these values through 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 value dates. Their count is taken as the answer for (f) or (g) depending on the class.
The overall runtime of completing $Task1$ is about $O(n)$ where n is the number of records in rows in the file.

2 Task2: Data pre-processing

For this section, I a built a list called *pattern* from the values of the partial data file. I also built a dictionary of the form $dic = \{local : \{indicator : [[dates], [values]]\}\}$ from the complex data file where local is a combination of *country and Locality*. This effectively grouped all the records in two different levels: localities and *Indicator* within each locality. Finally, each *Indicator* contains two lists, one for the dates and their respective cumulative values.

2.1 Main Task

I solved *Task2* by employing the Knuth-Morris-Pratt(KMP) pattern search algorithm(Mandumula, 2011). I built the suffix list from the pattern built in Data pre-processing above. With that suffix, I iteratively searched for the pattern in all $values_{ij}$ where $values_{ij} = dic[local_i][indicator_{ij}][1]$ if the the size of $values_{ij} \geq$ the size of the $pattern$. If the $pattern$ is found, *local, indicator, start_date* are returned, where *start_date* = $dic[local][indicator][0][kmpIndex]$ and *kmpIndex* = index of the pattern in $values_{ij}$. The runtime is $O(mn(n + k))$ Where m is the number of distinct localities, n is the number of entries per indicator and k is the size of the pattern such that $n \geq k$

References

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