Data Structure & Algorithms Project

Phase 2

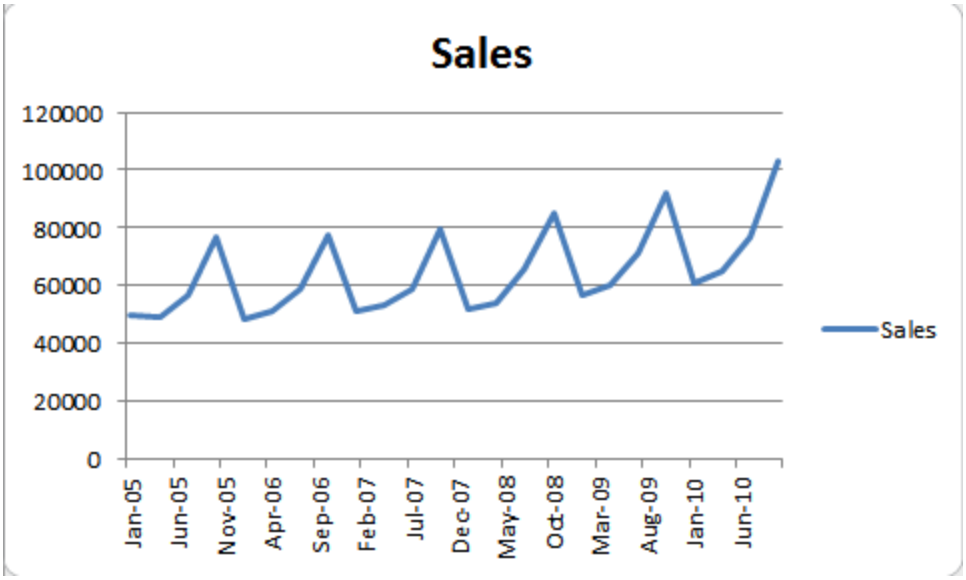
- In this phase (2) you'll implement an algorithm to identify motifs several signals.

Prerequisites:

1. Time Series:

   A time series $T =< t_1, t_2, \dots, t_{n-1}, t_n >$ is a series of data points (t_i) indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time.

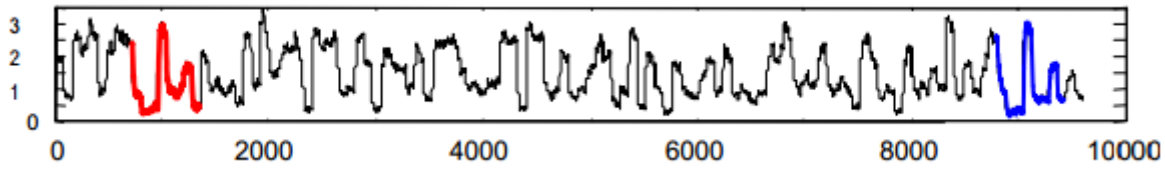
   Example: a time series showing the amount of sales for a product



2. Time Series Motifs:

   Time series motif $M_j =< t_i, t_{i+1}, \dots, t_{i+m} >$ consists of a temporally ordered subsequence of a time series with length m on dimension j.

Example:



- **Your Tasks**:
  - discretize data using given SAX algorithm (python and java libraries are attached). for example a signals may be converted to the following string:

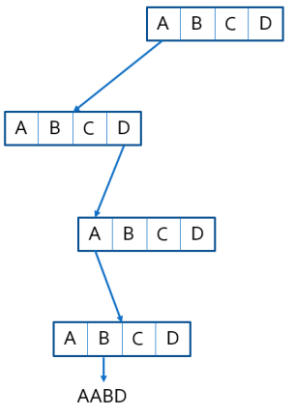
    Signal = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,6,6,6,6,10,100]

    String with word size 6 and alphabet size 5: b b b b c e
    (for threshold you can give a very small number like 10^{-6})

    Also extract all letters from subsequence as alphabet. For the above example, alphabet letters are A, B, C.

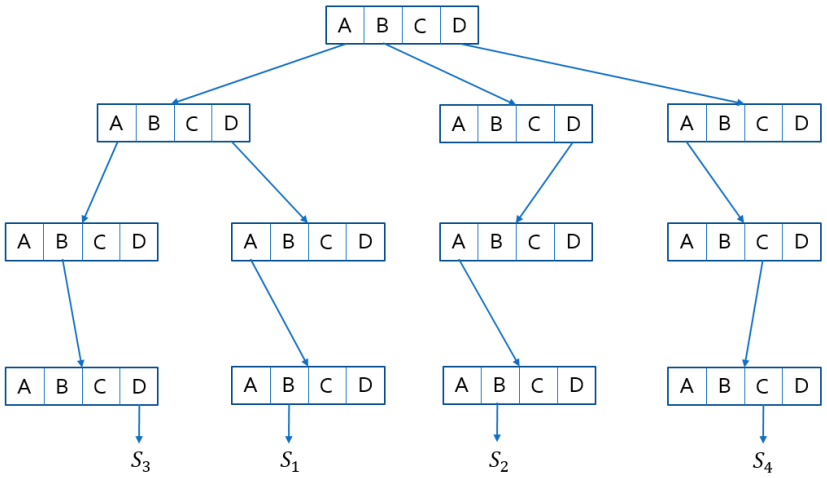
  - After executing SAX algorithm, you must implement your own tri data structure and algorithms, without using **any** additional libraries. Save every subsequence in the tri.

    You must build the tri as follows:
    - Each node consists of all **used** alphabet letters m in subsequences as data , and m pointer
    - To traverse the string you must interleave elements from the beginning and the end of the string, i.e, the first symbol, then the last symbol, then the second and so on → ABCDEF → A – F – B – E – C – D
    - Example of tree after inseting AABD, the alphabet is A,B,C,D. Note that the letter insetion order is A, D, A, B:

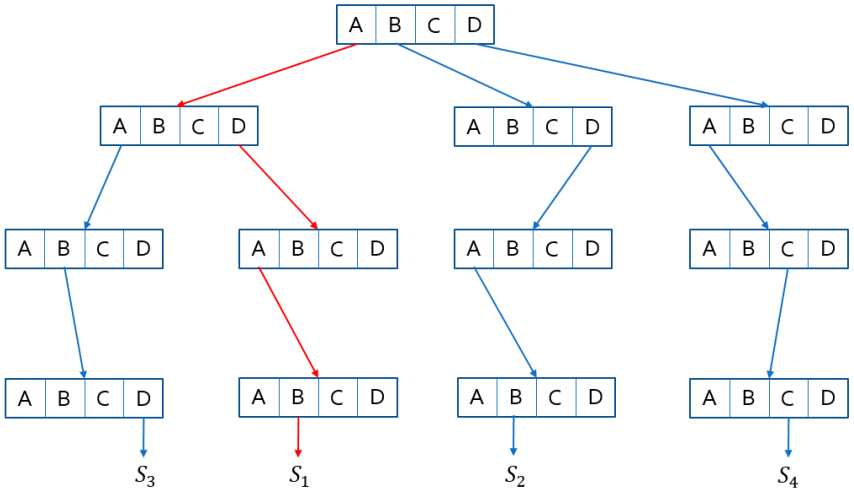


- You can see the full example here after inserting the following subsequences:

$$S_1 = AABD \mid S_2 = BABD \mid S_3 = ABDA \mid S_4 = DCCA$$

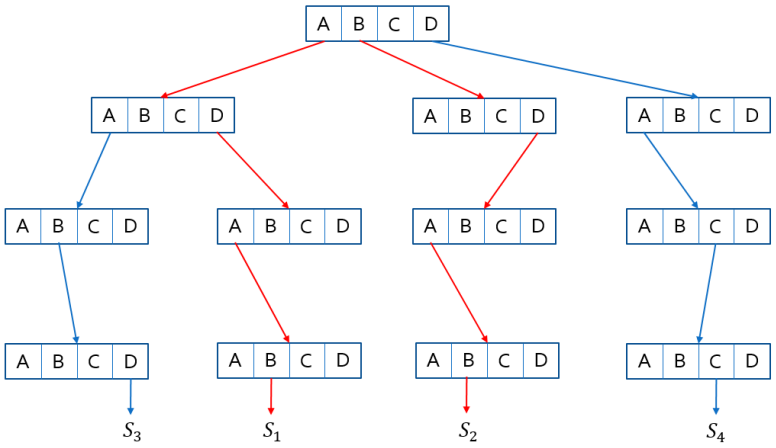


- After building such tri on the given dataset, your program must be able to get a subsequence as input (from user or file), and finds all motifs for the input, to find motifs you must traverse several paths of the tri:
  - Finding motifs for AABD:
  1. First letter is A → going down A pointer
  2. Next (last) letter is D → going down D pointer
  3. Next (second) letter is A → going down A pointer
  4. Next (third) letter is B → we've reached a leaf, B pointer points to S_1, so S_1 is a motif for AABD (they are equal!)



- The algorithm didn't find BABD as an answer, to overcome this problem, you must consider a margin δ, and traverse each path within range $|current\ letter - Node_{symbol}| \leq \delta$. That is, if $current\ letter = B$ and $\delta = 1$, the algorithm visits the paths of A, B and C (one up or one down).
- Finding motifs for AABD again with $\delta = 1$:

1. First letter is A → we must check A and B (A + 1), going down A and B pointer, first we check A, after reaching a leaf, we'll go for B pointer.
2. Next letter is D, we should check C and D pointers. C is NULL, so we only check D pointer.
3. Next letter is A, we should check A and B pointers, B is NULL, so we only check A pointer.
4. We've reached a leaf, next letter is B, S_1 is found.
5. We go back to B pointer at top and continue till we reach S_2.



- Consider given dataset file named 'dataset.txt', this dataset consists of 500 data. Each line is a single data consists of 200 to 500 numbers. Convert the whole

dataset into strings and build a tri for your dataset. Then give your program some inputs and check the outputs. You must test your program with word size 6, 7 and 8, with alphabet size 10. Your code must **flexible** to parameter changes (changing word size, alphabet size and delta).

- Calculate time for building tri and finding motifs for 1 subsequence.
- What does the SAX algorithm do? Explain.

Bonus: Implement you own SAX algorithm to convert a signal to a string with size w and alphabet size n.

Libraries:

SAX Algorithm:

- SAX Python: https://github.com/nphoff/saxpy : make an instance from SAX class and use to_letter_rep(word_size, alphabet_size, threshold) function.
- SAX Java: https://github.com/jMotif/SAX
- SAX C++: https://github.com/melsabagh/sax

Notes:

- Your implementation should be functional
- You report must be in Persian without copying any paper's text
- Any sign of cheating will result in the **zero** grade, your codes logic will be checked automatically with your classmates codes and codes from the web
- Your report should be a single PDF file containing answer to questions and your results, include two or three sample input and outputs (your own program output) in your report
- You should upload your codes and report in a single ZIP file named 'STD ID 1 – STD ID 2.zip' (e.g. '902717 – 9433894.zip')

GoodLuck ☺