**Question 1: Swimming Pools**

My house has a open swimming pool in the backyard. The water in the swimming pool keeps on diminishing during the day because of sunlight. During hot days, the water evaporates faster, but during humid days the water evaporates slower. During cold days, the water evaporates very little. I want to build an autonomous pump which is continuously pumping water into the pool so that the water level in the pool does not change (it does not go up or go down).

I take a simple pump and connect it with a computer that maintains the flow of water through the pump. I also have a weather meter that provides readings of temperature and humidity every minute. I also have a water level sensor which provides a reading every minute.

* Design a model, that tracks the rate of change of water level in the swimming pool and makes continuous changes to water pump. How would estimate a model to solve for this problem?
* Since the water in pool is hard water, it causes scaling in the pump over time. This means that the pump performance tends to change over time. How will your model adapt to this change in pump performance automatically?

Feel free to use concepts from differential equations, dynamics, machine learning and statistics to come up with a solution to this problem.

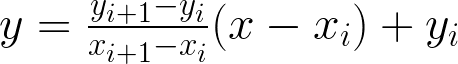
**Question 2: LINEAR SPLINES**

You might have noticed linear interpolation in chart plotting softwares like Excel , where when you plot the data-set using the line chart, then the points are joined together by line segments as shown below. This is done using linear splines.

Chart, line chart, scatter chart

Description automatically generated

Linear splines are created using a pair of adjacent points to be plotted in a chart. Assuming you have two points   (x_i,y_i)     and (x_{i+1},y_{i+1}) , the equation associated with the spline for those two points is given by:



How would you write code in any language of choice to solve the following?

1. Ask the user for a set of 20,000,000 points (x,y) to connect with a piecewise spline that connects all these data points in the sequence in which they were received.
2. Use the set of points the user has provided and sort them based on the x values. Then create a piecewise spline that connects all these data points. (Hint: Think about the sort algorithm you are going to use.)
3. Given that this code is going to estimate piecewise splines on a very large number of data points, explain techniques you would use to improve the execution of your code and reduce compute time? Would these techniques differ between the point 1 and point 2 above?

**Question 3: Matrix Operations**

1. Consider a data structure to represent a matrix of dimension 1,000,000 x 1,000,000.
   1. How would you design a data structure appropriate for a dense matrix and a sparse matrix respectively?
   2. Clearly explain how you will determine whether the matrix should be supported using a sparse matrix or a dense matrix?
   3. Explain how you will store the matrix - disk or in-memory or a hybrid approach.
2. How would you calculate the rank of the matrix and eigenvalues of the matrix?
3. Write your point of view on the type of changes you would make to these algorithms for such a large matrix size. (At the current size, traditional algorithms become really slow and bogged down)

**Hints**

**How to calculate the Rank of a matrix?**

1. Let the input matrix be mat[ ][ ].  Initialize rank equals to number of columns
2. Do the following for each row k = 0 to rank-1.

* If mat[k][k] is not zero, make all elements of  current column as 0 except the element mat[k][k]  by finding appropriate multiplier and adding a the  multiple of row 'k'
* Else (mat[row][row] is zero). Two cases arise:
  + If there is a row below it with non-zero entry in the same column, then swap current row ‘k’ and that row.
  + If all elements in the current column below mat[r][k] are 0, then remove this column by swapping it with the last column and  reducing the number of rank by 1.Reduce row by 1 so that this row is processed again.

1. Number of remaining columns is the rank of the matrix.

**How to calculate the eigenvalues of a matrix?**

1. Read Order of Matrix (k) and Tolerable Error (e)
2. Read Matrix mat of Size k x k
3. Read Initial Guess Vector X of Size k x 1
4. Initialize: Lambda\_Old = 1
5. Multiply: X\_NEW = A \* X
6. Replace X by X\_NEW
7. Find Largest Element (Lamda\_New) by Magnitude from X\_NEW
8. Normalize or Divide X by Lamda\_New
9. Display Lamda\_New and X
10. If |Lambda\_Old - Lamda\_New| > e then set Lambda\_Old = Lamda\_New and goto step 5 otherwise report X.