Analysis of Mark Sort Algorithm for Database column swaps

```c
void mrkSort(Array2D *a,int col){
    for(int i=0;i<a->rows-1;i++){
        for(int j=i+1;j<a->rows;j++){
            if(a->data[i][col]>a->data[j][col]){
                for(int k=0;k<a->cols;k++){
                    int temp=a->data[i][k];
                    a->data[i][k]=a->data[j][k];
                    a->data[j][k]=temp;
                }
            }
        }
    }
}
```

Equation representing the code ($O \rightarrow$ Operations)

$O_o \rightarrow$ Operations outer loop

$O_i \rightarrow$ " inner loop

$O_{if} \rightarrow$ " after if

$O_s \rightarrow$ " swap

R = Row, C = col, i, j = indexes

then

$$O + \sum_{i=0}^{R-2} \left(O_o + \sum_{j=i+1}^{R-1} \underbrace{\left(O_i + P\left(O_{if} + \sum_{k=0}^{C-1} O_s\right)\right)}\right)$$

Let $\quad O_i + P(O_{if} + C O_s) = S$

Substituting

$$O + \sum_{i=0}^{R-2} \left(O_o + \sum_{j=i+1}^{R-1} S\right)$$

Next round of simplifications with goal to obtain a function dependent on R

$$f(R) = C_0 + C_1 R + C_2 R^2$$

to continue

$$f(R) = O + \sum_{i=0}^{R-2} \underbrace{\left(O_0 + \sum_{j=i+1}^{R-1} s\right)}$$

$$O_0 + ((R-1) - (i+1) + 1)s$$

$$O_0 + (R-1)s - is$$

for the next Loop

$$f(R) = O + \sum_{i=0}^{(R-2)} \left(O_0 + (R-1)s - is\right)$$

$$= O + (R-1)O_0 + (R-1)^2 s - \frac{(R-2)(R-1)}{2} s$$

which simplifies to

$$f(R) = (O - O_0) + (O_0 - s/2)R + (s/2)R^2$$

There fore

$$C_0 = O - O_0$$

$$C_1 = O - s/2 \qquad \text{where } s = O_i + P(O_{if} + CO_s)$$

$$C_2 = s/2$$

finally

$$f(R) = C_0 + C_1 R + C_2 R^2$$

3/8

Q.E.D.

What does this say about how this algorithm scales, i.e. $R \to \infty$

$$f(R) = O(g(R))$$

$$= \{ f(R) : R, R_0 \in Z, c \in R; \quad 0 \le f(R) \le c\, g(R) \ \forall R > R_0 \}$$

Let $g(R) = R^2$

then $$0 \le C_0 + C_1 R + C_2 R^2 \le C R^2$$

$$0 \le \frac{C_0 + C_1 R + C_2 R^2}{R^2} \le C$$

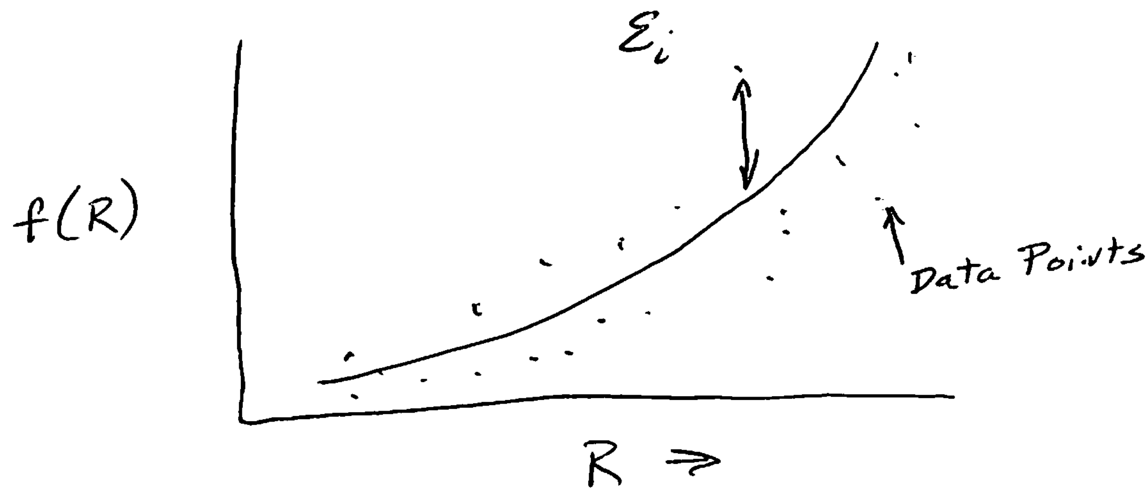
$$0 \le \frac{C_0}{R^2} + \frac{C_1}{R} + C_2 \le C$$

$$\lim_{R \to \infty} 0 \le \cancel{\frac{C_0}{R^2}}^{\,0} + \cancel{\frac{C_1}{R}}^{\,0} + C_2 \le C$$

then $$f(R) = O(R^2)$$

Q.E.D.

Derivation of Least Squares Curve Fit for empirical data derived from timing or operational analysis.



Let

$m \rightarrow$ Number of data points

$N \rightarrow$ Number of coefficients to solve for from empirical data

$$m > N$$

$\underset{m \times N}{R} \rightarrow$ Matrix relating size or number of rows run vs. the function to curve fit.

$\underset{m \times 1}{\underline{f}} \rightarrow$ Vector containing the empirical number of operations or time algorithm took to run.

$\underset{m \times 1}{\underline{\mathcal{E}}} \rightarrow$ Vector representing error between curve to fit and actual data points

$\underset{N \times 1}{\underline{c}} \rightarrow$ Proposed Vector of coefficients to find relating to the type of function.

Equation which defines the curve your trying to fit with it's associate errors.

$$\underset{M\times 1}{\underline{f}} = \underset{M\times N}{R}\,\underset{N\times 1}{\underline{C}} + \underset{M\times 1}{\underline{\varepsilon}}$$

Goal: Minimize the errors to find the best curve fit.

Rearranging in terms of the errors

$$\underline{\varepsilon} = \underline{f} - R\underline{C}$$

Square the error which becomes our cost function

$$J(\underline{c}) \Rightarrow \underset{1\times m}{\underline{\varepsilon}^T}\,\underset{m\times 1}{\underline{\varepsilon}} = (\underline{f} - R\underline{C})^T(\underline{f} - R\underline{C})$$

$T \rightarrow$ represents the transpose

$$= \underline{f}^T\underline{f} - \underline{f}^T R\underline{C} - \underline{C}^T \underline{R}^T \underline{f} + \underline{C}^T R^T R\underline{C}$$

$$= \underline{f}^T\underline{f} - 2\underline{f}^T R\underline{C} + \underline{C}^T R^T R\underline{C}$$

Note:

$$J(\underline{c}) = \underline{f}^T\underline{f} - 2\underline{f}^T R\underline{c} + \underline{c}^T R^T R\,\underline{c}$$

is the same as

$$= \underline{f}^T\underline{f} - 2\underline{c}^T R^T\underline{f} + \underline{c}^T R^T R\,\underline{c}$$

We would like to minimize the cost function by taking the derivative and setting equal to the zero vector

$$\underline{\phi} = \frac{\partial(J(\underline{c}))}{\partial\,\underline{c}} = \frac{\partial\left(\underline{f}^T\underline{f} - 2\underline{c}^T R^T\underline{f} + \underline{c}^T R^T R\underline{c}\right)}{\partial\,\underline{c}}$$

$$\underbrace{\underline{\phi}}_{N\times 1} = -\,2\underbrace{\underset{N\times M}{R^T}\ \underset{M\times 1}{\underline{f}}}_{N\times 1} + 2\underbrace{\underset{N\times M}{R^T}\ \underset{M\times N}{R}\ \underset{N\times 1}{\underline{c}}}_{N\times 1}$$

Which gives

$$R^T R\,\underline{c} = R^T\underline{f}$$

and finally

$$\underline{c} = (R^T R)^{-1} R^T\underline{f}$$

Q.E.D.