Assignment 1

> Author: Isaac Archer
>
> Name: Assignment 1, Finding Nearby Differences
>
> Due Date: Wed, July 1st, 2015 00:00

Section 1: Formal Problem Description

> problem name: find maximum nearby differences
>
> instance: array of integers
>
> problem solution: $\exists$ (i, j) $\in \mathbb{Z}$ {1..n} | $\forall$ (k, l) $\in \mathbb{Z}$ {1..n} | d(i, j) > d(k, l)

Section 2: Algorithm

> algorithm
>
> - input: an array of integers of size n, nums
> - output: i and j for maximum d(i,j) <- |a[i] - a[j]| / |i-j|

```
algo <- nums

    max_i <- 1
    max_j <- 2
    max_d <- 0
    cur_d <- 0

    for i <- 1..(length(nums) - 1)
        for j <- (i+1)..length(nums)

            cur_d <- d (i, j, nums)

            if cur_d > max_d
                max_i <- i
                max_j <- j
                max_d <- cur_d

    return max_i, max_j
```

> definition for d(i, j)
>
> - input: two integers i, j and an array of integers, nums
> - output: d(i,j) <- |a[i] - a[j]| / |i-j|

```
d <- i, j, nums

    temp_a <- |nums[i] - nums[j]|
    temp_b <- |i - j|

    return temp_a / temp_b
```

Section 3: Complexity

> - assuming the basic operation is the number of calls to d(i, j) we have complexity t[n] = ∑ i for (i..n)
> - this simplifies: t[n] = n(n + 1) / 2

- this simplifies: t[n] = (n^2 + 2) / 2
- this simplifies: O(n^2)
- the overall complexity is a gaussian sum but the complexity class is not better than O(n^2).
- my algorithm is in the space of n^2 or O(n^2)

Section 4: Correctness

> the double nested loop of for(i [1:n]) { for(j [1:n]) { } } meets the requirement to check all possible combinations of a[i][j]. this allows me to check all d(i, j) values and compare them to previous d(i, j values) using the transitivity of of greater than and find the highest d(i, j) for the array of integers a. i store the current i and j values for the current highest d(i, j) and then return this at the end. they are only updated when d(i, j) exceeds the previous d(i, j). this shows that the algorithm gives the correct i and j values for the max d(i, j) for the integer array a.