**Design Document**

**Assignment 2 PS8 : Bitcoin Investment Application**

**Group : Group 128**

**Members**:

Vinutha : 2019AB04217

Nayaz Basha H M : 2019AB04163

Dileep R Dominic : 2019AB04015

M R Rahamathulla : 2019AB04229

Below explain the 2 approaches for solving the bitcoin investment problem. Our aim is to maximize the profit for the given month by having one buy and one sell in a month.

The two approaches ie divide and conquer approach with a recursive algorithm and a non recursive linear approach is explained in the document.

**Basic structure of python solution**

The solution to problem is written in python 3.7 . The input is read from data/inputPS8.txt

The class "bitcoinprofit” contains two functions “DivideAndConquerProfit()” and “LinearMaxProfit()” which are the two different approaches.

Once the solution is calculated the output is written back to data/outputPS8.txt as per the give guidance.

**Divide and Conquer strategy**

Function : DivideAndConquerProfit()

Here we split the array into two (roughly equal) halves. Then there are 3 possibilities to have a solution. If we do so, then there are three possible options about where the best buy and sell times are:

1. Buy and sell purely in the left half of the list.

2. Buy and sell purely in the right half of the list.

3. Buy in the left half of the list and sell in the right half of

the list.

One point to note is we need not consider selling in the left half of the list and buying in the right half of the list, since the buy time must always come before the sell time.

The case of 1 and 2 can be done by recursively invoking the function.

Now for case 3 alone as mentioned above we need to consider the below approach.

If the list has size 0 or size 1, the maximum profit is 0.

Else

i) Split the list in half.

ii) Compute the maximum single-sell profit in the left list, call it LEFT.

iii) Compute the maximum single-sell profit in the right list, call it RIGHT.

iv) Find the minimum of the first half of the list, call it MIN

v) Find the maximum of the second half of the list, call it MAX

vi) Return the maximum of LEFT, RIGHT, and MAX - MIN.

**Divide and Conquer strategy - Analysis**

Let's consider the time complexity of this algorithm. The base case takes O(1) time, and in our recursive step we make two recursive calls, one on each half of the array, and then does O(n) work to scan the array elements to find the minimum and maximum values. So recurrence relation is

T(1) = O(1)

T(n / 2) = 2T(n / 2) + O(n)

The relation can be solved to O(nlogn)

**Linear Strategy**

Function : LinearMaxProfit()

This strategy is also a brute force strategy to solve the problem. We will just consider all pairs of values, and then pick the one with the highest net profit

**Linear Strategy Analysis :**

Linear Strategy Analysis in worst case can be written in the following form as its a brute force way to find the solution from a linear array.

n + (n - 1) + (n - 2) + ... + 1 = n(n + 1)/2

So the time complexity can be derived to O(n2) time