**Assignment 5**

Laleh Rostami Hosoori A01772483

**Program #1:**

**C-** Measure the run time of each algorithm (in A, and B) solving that set of problems for each problem size and plot a log/linear graph of "problem size" (x axis linear) vs. "total time to solve problems set" (y axis log). Both these algorithms will be slow! Problem sizes will be small.

**log/linear Plot**

**D-** What is the run time as a function of problem size? Use (f(n) to express the function complexity where n is the problem size)

**AnotherBruteForcePDP**:

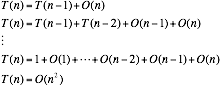
This algorithm examines  different sets of integers, but , so AnotherBruteForcePDP takes roughly  time.

**PartialDigest**:

Let T(n) be the maximum time the algorithm takes to find the solution for an *n*-point instance of the PDP. If there’s only one viable alternative



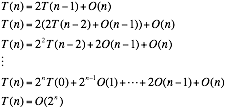
O(n) is the time spend adjusting the sets X and L. This case is quadratic.



But when there are two alternatives:



The runtime will be exponential.



**Program#2:**

**C-** Measure the run time of each algorithm (in A, and B) solving that set of problems for each problem size and plot a log/linear graph of "problem size" (x axis linear) vs. "total time to solve problems set" (y axis log). Both these algorithms will be slow! Problem sizes will be small.

**log/linear Plot**

**D-** What is the run time as a function of problem size? Use (f(n) to express the function complexity where n is the problem size)

**BruteForceMofitSearch**:

The number of sequences is  which is exponential in *t*. For each sequence, the algorithm calculates the score, which requires *O(l)* operations. So the overall complexity of the algorithm is evaluated as 

**BruteForceMedianSearch**:

It considers each of 4l strings of length *l* and computes TotalDistance at every step which spends *O(nt)*. Therefore, it has running time .