# Week 4 Lab 3: Algorithm Analysis

# 

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 2-sum

Using Raptor, Input 6 integers and store them in an array. The numbers can be hard coded, assigned by input or assigned at random (Random \* 100 in raptor)

Design an algorithm that calculates how many pairs of the 6 numbers sum exactly to zero. ie. if the numbers input were *5,10,20,-15,-10,20*, then this would produce 2 pairs that sum to 0:-

10 + -10 = 0

20 + -20 = 0

What would you have to change to make this a "3 sum" algorithm ie where 3 numbers summed to 0?

|  |  |
| --- | --- |
| Raptor code : | |
| Declare an array | Array\_values[1000] |
| Assign an array element | Array\_values[1]=20 |
| A random number in the range 1-100 | Random \* 100 |
| Timing your code | there is a system variable called Current\_Time. Assign this to variables at the start and end of an operation to time it. |

## Implement the above in C

srand((unsigned) time(NULL)); // mix up the random number generator

randomArray[i]= (rand() % N)+1 // generate a random number and assign it.

e.g. for (int i=0;i<N;i++) a[i]=(rand()%((N+1)-1))+1;

timing – #include <time.h> clock\_t startTime=clock();

## Chess

According to a well-known legend, the game of chess was invented many centuries ago in northwestern India by a certain sage. When he took his invention to his king, the king liked the game so much that he offered the inventor any reward he wanted.

The inventor asked for some grain to be obtained as follows: just a single grain of rice was to be placed on the first square of the chess board, two on the second, four on the third, eight on the fourth, and so on, until all 64 squares had been filled.

If it took just 1 second to count each grain, how long would it take to count all the grain due to him?



Using Raptor, develop a flowchart which calculates the answer