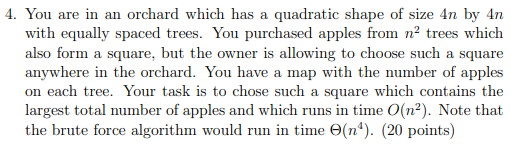
**COMP3121 ASSIGNMENT1 QUESTION4**

****

**Answer:**

We can see that the orchard is a quadratic shape of size 4n by 4n.

We can use x axis(J) and y axis(K) for the location for each tree. Length should be1… 4n.call A.

First we create array B [J][K] to store the sum of the apples of in the rectangle from [1][1] tree to [J][K] tree. In that case we should go through all the tree in this orchard which in time.

To do that we may have four case to consider.

For the [1][1] we just make B[1][1] = A [1][1]

For the 1st row, except [1][1] it only count the B[J][1] = B [J-1][1] + A[J][1]

For the 1st column, except [1][1] it only count the B[1][K] = B [1][K -1] + A[1][K]

For the other, it will count B[J][k] = A[J][K] + B[J - 1][K] + B[J][K -1 ] - B[J - 1][K - 1]

Because it always take in O(1) and we should go through 4n2 the total time should be .

After that we should have array B which numbers of apple in each rectangle from[1][1] to [J][K]. For the biggest number of apple square in the orchard, we have to check all the possible square and find the biggest apples square. So we have to go through array B [J][K] for all possible square. We can easily get the number of apple of n2 square using B[J][K] - B[J][K - n] - B[J - n][K] + B[J-n][K-n] and it will do in O(1). N J 4N, NK4N

Because we check B [J][K] and make compute the apple of square and choose bigger every time. It will take

Then we use biggest number to go through B [J][K] and make compute the apple of square to find the J and K we can know the location of square. (J, K), (J-N+1, K), (J, K-N+1), (J-N+1, K-N+1).