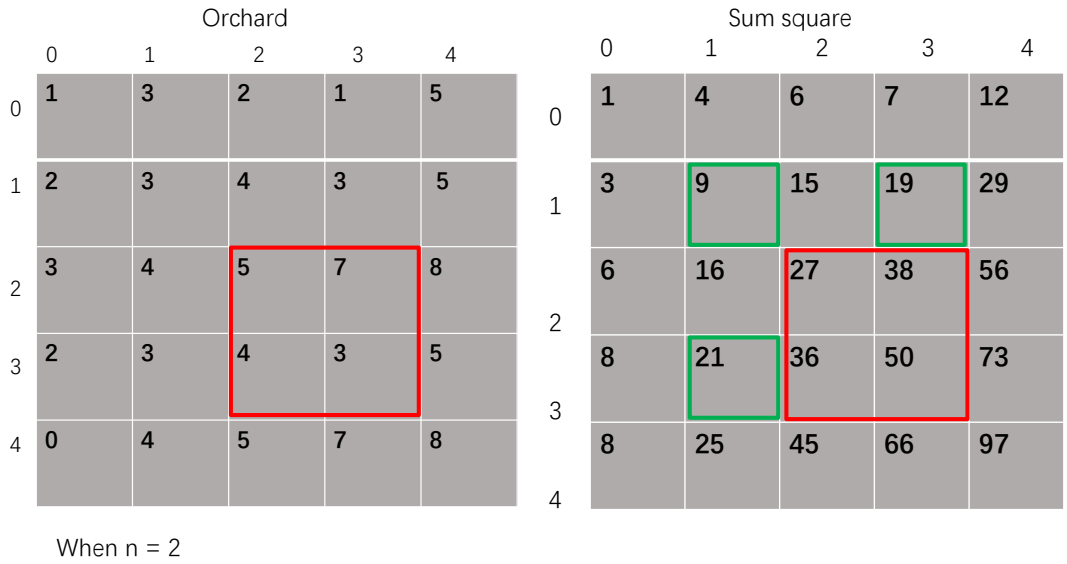
Yuan Gao z5239220 Q4

You are in an orchard which has a quadratic shape of size 4n by 4n with equally spaced trees. You purchased apples from n2 trees which also form a square, but the owner is allowing to choose such a square anywhere in the orchard. You have a map with the number of apples on each tree. Your task is to choose such a square which contains the largest total number of apples and which runs in time O(n^2). Note that the brute force algorithm would run in time Θ(n^2). (20 points)

Let's assume:



According to this graph:

The value of the element in the lower right corner $Or[i][j]$

$$Or[i][j] := Or[i][j] + Or[i-1][j] + Or[i][j-1] - Or[i-1][j-1]$$

Through this formula, the left side Sum square can be created.

The sum of n^2 square:

$$Sum = Or[i][j] - Or[i-n][j] - Or[i][j-n] + Or[i-n][j-n]$$

Therefore, we only need to traverse once to find the maximum value

The Time complexity is $O(n^2)$

```cpp
/*the Two-dimensional vector reprsent orchard,
  n as the length of purchased square
  ii as the maximum i index
  jj as the maxinum j index */
int getMaxApple(vector<vector<int>> & orchard, int n, int & ii, int & jj){
    int max = -1;
    int tmpMax = 0;
    for(int i = 1; i < 4 * n; ++i){
        orchard[i][0] += orchard[i - 1][0];
        orchard[0][i] += orchard[0][i - 1];
    }
    for(int i = 1; i < 4 * n; ++i){
        for(int j = 1; j < 4 * n; ++j){
            /*The value of the element in the lower right corner*/
            orchard[i][j] += (orchard[i - 1][j] + orchard[i][j - 1]
                              - orchard[i - 1][j - 1]);
            if(i >= n && j >= n){
                /*get the sum of square*/
                int tmpMax = orchard[i][j] - orchard[i - n][j]
                             - orchard[i][j - n] + orchard[i - n][j - n];
                if(tmpMax > max){
                    max = tmpMax;
                    ii = i;
                    jj = j;
                }
            }
        }
    }
    return max;
```