Assignment 2

Avijit Ghosh 14CH3FP18

January 2016

1 Triominoes

1. Place empty square randomly on any nxn checkboard, where n=power of 2 form.

2. Recursive Case:

Check in which quadrant the the empty square lies. Fill the other 3 centre squares with a counter, thus making it one triomino.

Eg, if my array is

```
-1 -1
0 -1
```

where 0 is empty, I will place a triomino on the three -1 to fill the centre. Also, square 2*2 is the base case.

3. One the centre triomino is placed, the square is divided into four sub squares on each of the four quadrants. The process is recursively repeated.

Example, for an array of size 4, my program outputs:

```
Enter size of array. It should be a power of 2:
4
3 3 5 5
3 1 1 5
4 1 0 6
4 4 6 6
```

2 Minimum Distance of two points on a plane

Input: An array of n points P[] Output: The smallest distance between two points in the given array.

1) Input array is sorted according to x coordinates. I take P[n/2] as middle point.

2) Divide the given array in two halves. The first subarray contains points from P[0] to P[n/2]. The second subarray contains points from P[n/2+1] to P[n-1].

3) Recursively find the smallest distances in both subarrays.The base case is when the number of points in any half is less than or equal to three, in which case we run a simple pairwise matching three times (brute force) and return the

minimum. Let the distances be dl and dr. Find the minimum of dl and dr. Let the minimum be d.

4) From above 3 steps, we have an upper bound d of minimum distance. Now we need to consider the pairs such that one point in pair is from left half and other is from right half. Consider the vertical line passing through passing through P[n/2] and find all points whose x coordinate is closer than d to the middle vertical line. Build an array of all such points in the zone and arrange the point according to y coordinates.

5) Find the smallest distance in the zone. It can be proved geometrically that for every point in zone, we only need to check at most 7 points after it by the pigeonhole principle (note that zone is sorted according to Y coordinate).

6) Finally return the minimum of d and distance calculated in above step.