

Report for assignment 2

Vasudha Todi (14EC10059)

January 13, 2016

Divide and Conquer

1. **To fill an $n \times n$ board with one missing tile using L shaped tiles (n is a power of 2)**

The problem is solved using divide and conquer by dividing the board into 4 parts in each recursive step. One of the 4 sub-squares will contain the missing tile. An L shaped tile is placed at the center using the remaining 3 sub-squares, which then act as the missing tiles in the corresponding sub-problems.

Algorithm 1 Divide and Conquer 1

```
1: Board( $A, n$ )
2: if  $n = 2$  then //Base case
3:   Put an L shaped tile in the remaining 3 boxes
4:   return
5: end if
6: Determine the sub-square containing the missing tile
7: place an L shaped tile at the center using the remaining 3 sub-squares.
8: Recursive call to A() for each sub-square
9: return
```

The tiles are displayed using different numbers.

Results :

Value of n : 4

Missing tile at : (2,3)

Output :

$$\begin{bmatrix} 2 & 2 & 3 & 3 \\ 2 & 1 & 1 & 3 \\ 4 & 1 & 5 & -1 \\ 4 & 4 & 5 & 5 \end{bmatrix}$$

2. **To find the closest pair out of n points**

The n points are divided into two parts. The closest pair is found out recursively in

the two halves and the shorter of the two is taken. A strip is taken at the center to compare the points of the two parts with each other in a minimum number of steps.

Algorithm 2 Divide and Conquer 2

```
1: shortestDist( $A, Ax, n, start, end$ )
2: if ( $end - start \leq 3$ ) then
3:   Find the closest pair using brute force
4:   return min
5: end if
6:  $mid = (start + end) / 2$ 
7:  $a = \text{shortestDist}(A, Ax, n, start, mid)$ 
8:  $b = \text{shortestDist}(A, Ax, n, mid+1, end)$ 
9:  $sd = \min(a, b)$ 
10: Take a strip having y co-ordinates (mid-sd) to (mid+sd)
11: Store the points in the strip in the order of their x co-ordinates(total c points)
12: for  $i = 1, c$  do
13:   Check the distance between all the point in the strip within an x distance
   of sd from the ith point
14:   if  $dist < sd$  then
15:      $sd = dist$ 
16:   end if
17: end for
18: return sd
```

Results :

No. of points : 4

Points : (1,2), (4,5), (6,7), (10,12)

Output :

The closest points are (4,5) and (6,7)

The shortest distance between them is : 2.828427