Assignment 4

2A>

This solution can be applied to any general problem which involves calculation of minimum number of objects that needs to be distributed in a rectangular area. We divide the area in to number of small areas and we try to distribute objects to guarantee that all the local area scale are covered. This in turn ensures that on a global scale it results in coverage of complete global rectangular area.

This is also similar to bucket sort where instead of sorting using buckets, we find all the elements which fall into the same bucket, we just hold onto the first element and ignore the rest. If we come across an element which does not come under this bucket we add a new bucket. We repeat this until we have covered all the elements. At the end we return the elements in all the buckets which gives optimal output.

2B>

Let X = { $x_1, x_2, x_3, x_4, \ldots\ldots x_n$} be an array which consists of the distance of the centers of the paintings from the start of their respective walls.

Panels of lights need to be installed in the hall way and each panel of light will provide light for a span of “m” feet.

Steps to solve :

1. Check if the input array x is empty if so then return empty integer array.
2. Else Sort the array x in the increasing order of their distances from start position of hall way using merge sort algorithm.
3. Create an array “light” of max size x.length to store light panel positions
4. Next we consider the first painting in the sorted array. Place a panel of light to cover m feet distance starting from the first painting .
5. Add the first light position (c1 + (0.5 * m)) where c1 is the center of first painting and m is the number of feet span that each light can cover.
6. Now we check if the next painting center is within the recently installed light panel span, if it is then we just consider the next painting else we add a new light panel at position ck + (0.5 *m) to cover the painting at center ck for some ck > c1 to cover the span ck+ m. increment light panel counter by one

7. We repeat the process of checking by considering next . we repeat the step 6 until we cover all the paintings in the hallway. By the end of this loop we will have all paintings light up by at-most one light panels. Array “light” will have all the light panel positions
8. Return the array “light”.

Pseudocode:

Given : An array x which consists of the position of centers of the paintings from the starting point of the hallway and an integer m which represents the span the light panel can cover.
Returns : one dimensional array which consists of light position in the hall way.

```
HALL_WAY(x[], m)
{
        light[]   is an array of  size x.length to store the position of light
        panels installed

        if ( x.length == 0) return light

        // mergesort the array x in increasing order
        mergesort(x)

        // initially light is installed to cover m feet starting  from first
        //painting
        lightSpan = x[0] + m
        light[0] = x[0] +(0.5 * m)
        j = 1
        for i = 1 to x.length-1
              if x[i] > lightSpan
                lightSpan = x[i] + m
                light[j] = x[i] + (m * 0.5)
                j++
        return light
}
```

2 C>

Correctness :

First we sort the centers of paintings in the increasing order of their distances from the start position of the hall way.

Let $C_{i\,j}$ be the set of light panels installed in the hall way in an optimal way to light all the paintings in the hall way between i and j, where i and j represents the distances from start position of hallway S.T. i > j. Paintings between the position i and j are all light up.

Let k + (m * 0.5) be the position at which light is installed in an optimal solution. Then we are left with two sub-problems i.e. to find the set of lights that are required to light all the paintings to the left of k and all the paintings to the right of k

The lights in C_{ij} be the optimal solution with light installed at k . This can be represented as follows
$|C_{ij}| = |C_{ik}| \cup k \cup |C_{kj}|$ Where C_{ik} and C_{kj} represents the optimal solution to the sub problem on left and right respectively.

So to find optimal solution we could use the optimal sub-structure
Let n be the total number of paintings

$$C[i, j] = \begin{cases} 0 & \text{if } n = 0 \\ \min\limits_{i<=k<=j} \{ C[i, k] \cup k \cup C[k, j] & \text{if } n \,!= 0 \end{cases}$$

Let C_{ij} be the optimal solution set consisting of light position. The length of this set is If we could find a solution C_{kj}` S.T. length ($C_{kj}$)> length ($C_{kj}$`|), then we could use C_{ik}` rather than C_{ik}. Then the resulting solution can be constructed as
length (C_{ik} U k U C_{kj}) > length(C_{ik} U k U C_{kj}`) = length(C_{ij})
S. T all the painting in the span i and j are light by exactly one light.
But this a contradiction to the assumption we made that C_{ij} is optimal

Let C_k be the count of the minimum number of lights that can be installed in the hall way to cover all the k painting making sure that each painting is light by only one light. $C_k = C_{k-1} + 1$ where C_{k-1} is the optimum count of lights to cover k-1 paintings
If we assume $C_m < C_k$ to be some arbitrary count of light which can cover all the paintings K in the hall way then we have $C_m + 1 < C_{k-1} + 1 = C_k$ which is a contradiction our assumption we made. Therefore C_k is the optimum count of lights installed in the hall way

Time Complexity :

Let n be the count of painting.
Step 2 takes nlogn time to sort the array x . Run time = nlogn * c1

Steps from 7 to 11 run for a maximum of n times (i.e. in worst if each of the n paintings require one light panel to be installed). Run time = n * c2

All the other steps take constant amount of time . Run time = C

Total run time = n * c2 + nlogn * c1+ C

⇨ Time Complexity T(n) = O(nlogn) , by ignoring all lower order terms

Space Complexity :

In this solution, two variables are used . One is used to store the count of lights installed i.e j and one to keep track of light span which keeps track of total distance that is covered so far. Each of these take constant space

One 1-dimensional array of size n where n is the number of paintings in the hall way. This takes n space
. Therefore total space = 1 + 1 + n

⇨ Space-Complexity = O(n)