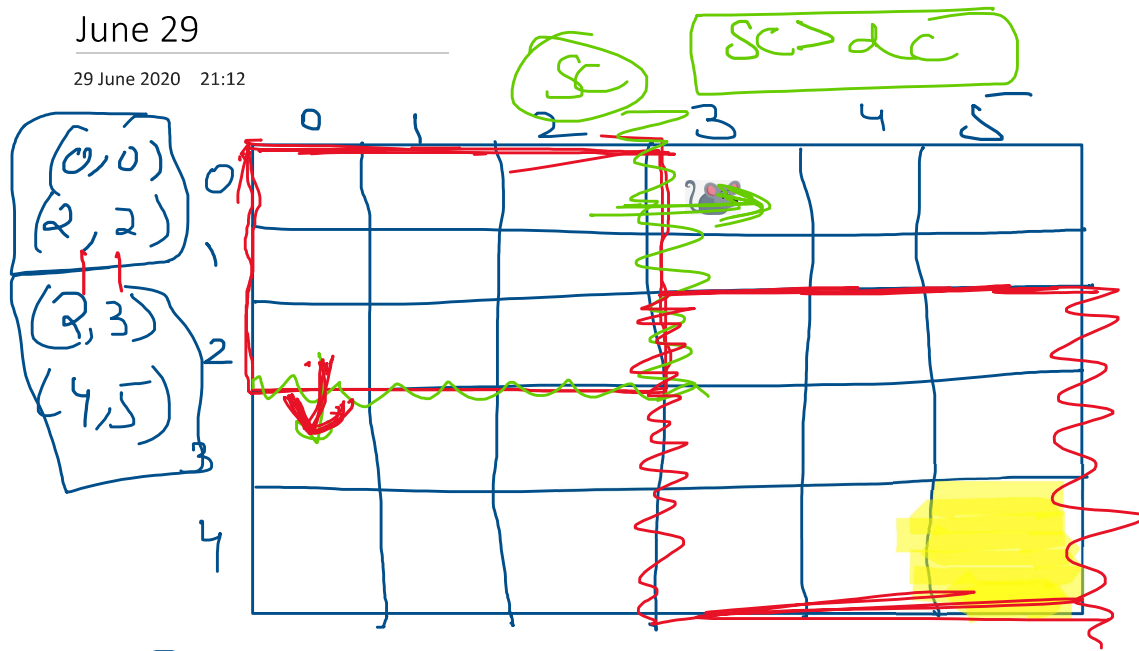


June 29

29 June 2020 21:12



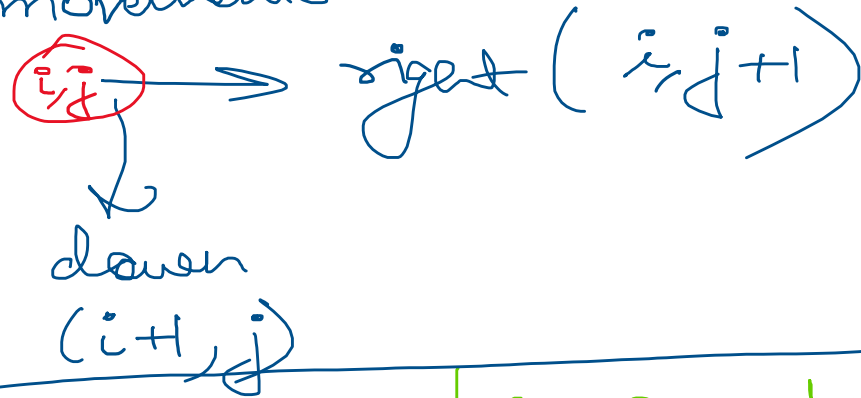
$$(2,2) (2,2) \rightarrow 1$$

$$\begin{matrix} m \times n \\ 5 \times 6 \end{matrix}$$

Source $\rightarrow (sr, sc)$

Destination $\rightarrow (dr, dc)$

Find the no. of ways
2 possible movements



Base Condition

findMazePath(sr, sc, dr, dc)

1. If $(sr == dr \text{ and } sc == dc)$

```

if (source == destination)
{
    return 1;
}

```

1 Path found

```

if (sc > dc || sr > dr)
{
    return 0;
}

```

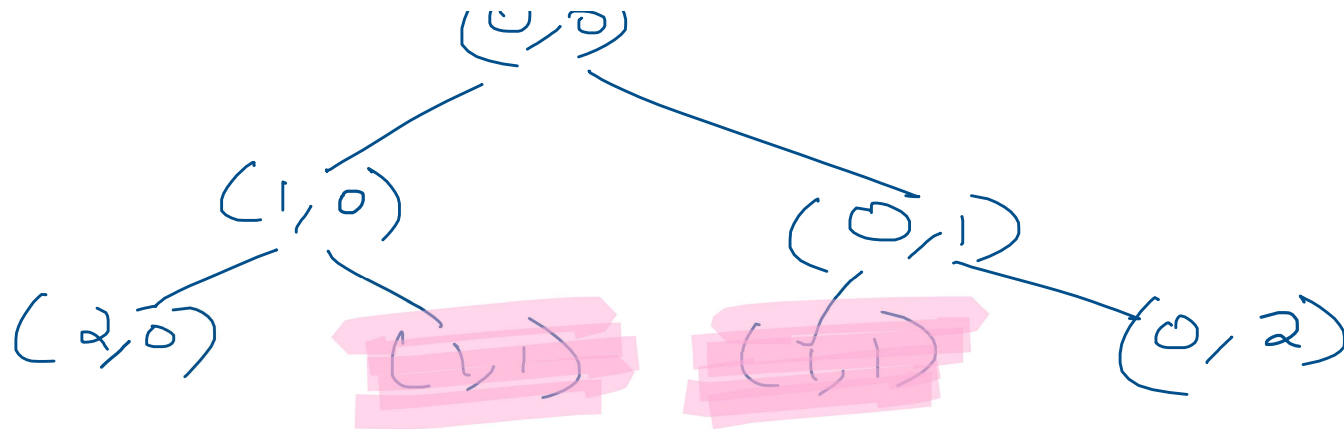
No. of ways = Right Take the ways +
 Left Take the ways.

```

return findMazePath(sr, sc+1, dr, dc)
+ findMazePath(sr+1, sc, dr, dc);

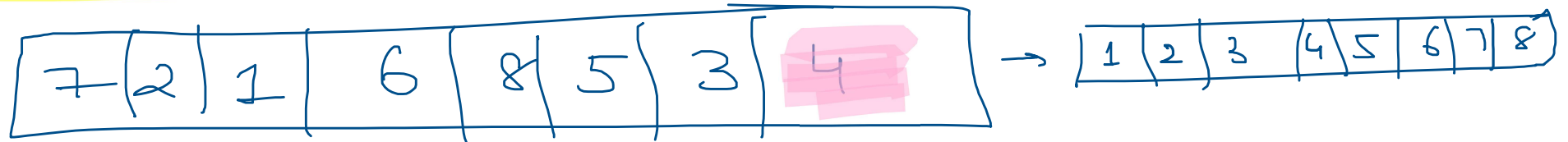
```

Recursion Tree



In Place Algorithm \rightarrow Requirement

Quick Sort to our Save



We will pick one Element \rightarrow Pivot

We will move elements smaller than it to the left
& the elements greater than it to the

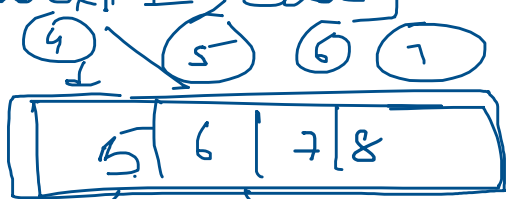
right.

\rightarrow 4 ko uski place pe sabh diya

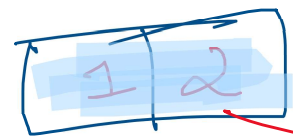
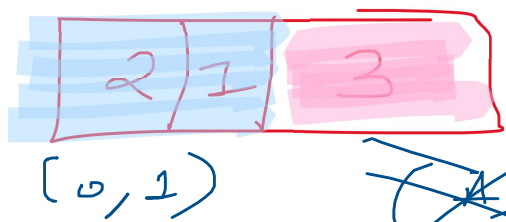


[0 / +]

[start, pIndex-1] pIndex [pIndex+1, end]



- Pivot Choose
- Partitioning
- Call QuickSort for the 2 arrays



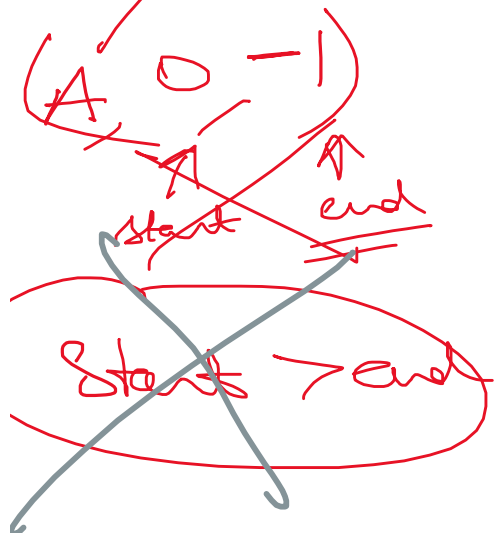
```

quickSort(A, start, end)
{
    if (start >= end) return;
    pIndex = Partitioning(A, start, end);
    quickSort(A, start, pIndex-1);
    quickSort(A, pIndex+1, end);
}
    
```

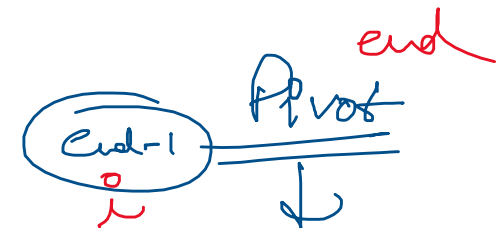
Same thing

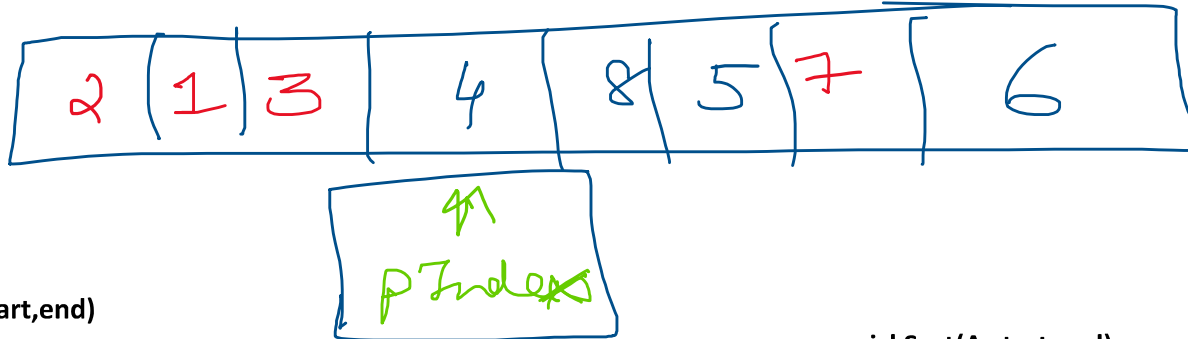
```

quickSort(A, start, end)
{
    if (start < end)
    {
        pIndex = Partitioning(A, start, end);
        quickSort(A, start, pIndex-1);
        quickSort(A, pIndex+1, end);
    }
}
    
```



Start





```

Partition(A,start,end)
{
    Pivot = A[end];
    pIndex = start;
    For(int i=start;i<end-1;i++)
    {
        If(A[i]<pivot)
        {
            Swap(A,i,pIndex); //swap values at index i && pIndex
            pIndex++;
        }
    }
    Swap(A,end,pIndex);
    return pIndex;
}

```

```

quickSort(A,start,end)
{
    if(start<end)
    {
        pIndex =
        Partitioning(A,start,end);
        quickSort(A,start,pIndex-1);
        quickSort(A,pIndex+1,end);
    }
}

```