# FINDING MAXIMUM AND MINIMUM 

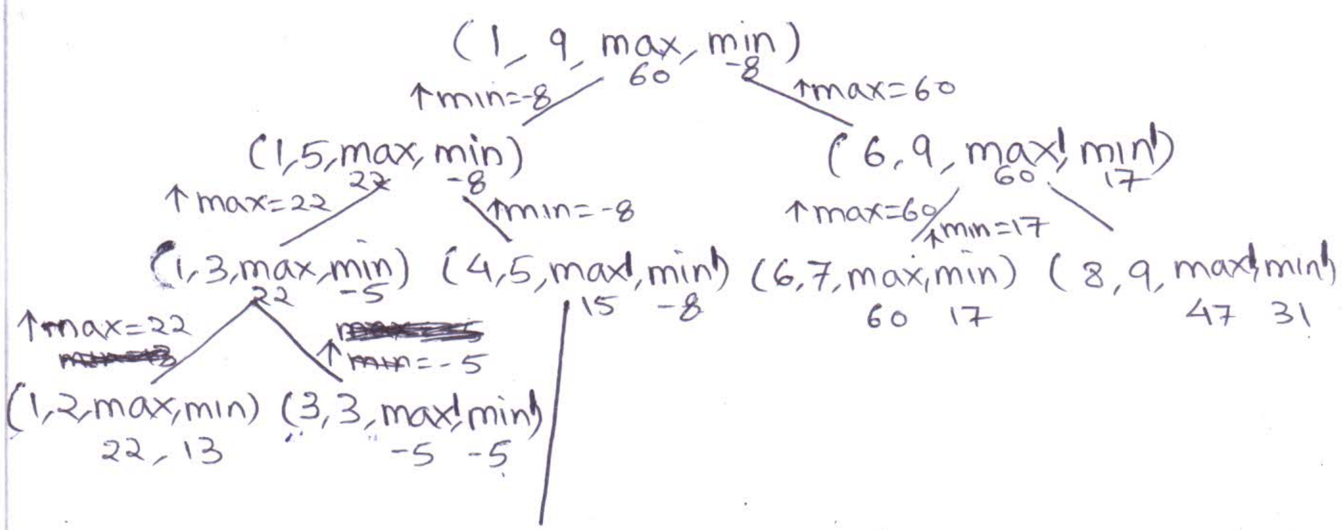
Following algorithm gives recursive (Divide-and-conquer) solution for finding maximum-and minimum problem.

```
MaxMin (i, j, max, min)
a[1..n] is a global array, parameters i and j
are integers, 1≤i≤j≤n, max and min
Stores the maximum and minimum elements
present in sub array.
{ if (i = j)        /* for single element */
  {
     max = min = a[i]
  }
  else if ( i = j-1)   /* for two elements */
  {
      if ( a[i] < a[j])
        {
           max = a[j], min = a[i]
        }
      else
        {
           max = a[i],  min = a[j]
        }
  }
  else
  {
        mid = ⌊(i+j)/2⌋     /* find mid of array */
        MaxMin(i, mid, max, min)  /* recursive call */
        MaxMin(mid+1, j, max1, min1)
        if (max < max1)
            { max = max1 }
        if (min > min1)
            { min = min1 }
} }
```

# FINDING MAXIMUM AND MINIMUM

a

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 22 | 13 | -5 | -8 | 15 | 60 | 17 | 31 | 47 |




max=60
min= -8

## Complexity Analysis

$$T(n) = \begin{cases} T(\lceil \frac{n}{2} \rceil) + T(\lceil \frac{n}{2} \rceil) + 2 & n > 2 \\ 1 & n = 2 \\ 0 & n = 1 \end{cases}$$

$$T(n) = 2T\left(\frac{n}{2}\right) + 2$$

$$= 2\left(2T\left(\frac{n}{4}\right) + 2\right) + 2$$

$$= 4T\left(\frac{n}{4}\right) + 4 + 2$$

$$\cdots$$

$$= 2^{k-1} T(2) + \sum_{1 \le i \le k-1} 2^i$$

$$= 2^{k-1} + 2^k - 2 = \frac{3n}{2} - 2$$

by Masters theorem

$$n^{\log_b^a} = n^{\log_2^2} = n$$

$$f(n) < n^{\log_b^a}$$

by case 1

$$\theta(n^{\log_b^a}) = \theta(n)$$