Finding Maximum and Minimum element using Divide and conquer

Iterative method

```
find max_min(A,n)
{ max = min = A[0];
  for(i=1; i<n; i++)   => n.
  { if(max<A[i])
        max = A[i]
    elseif (min>A[i])
        min = A[i]
  }
  return (max, min);
}
```

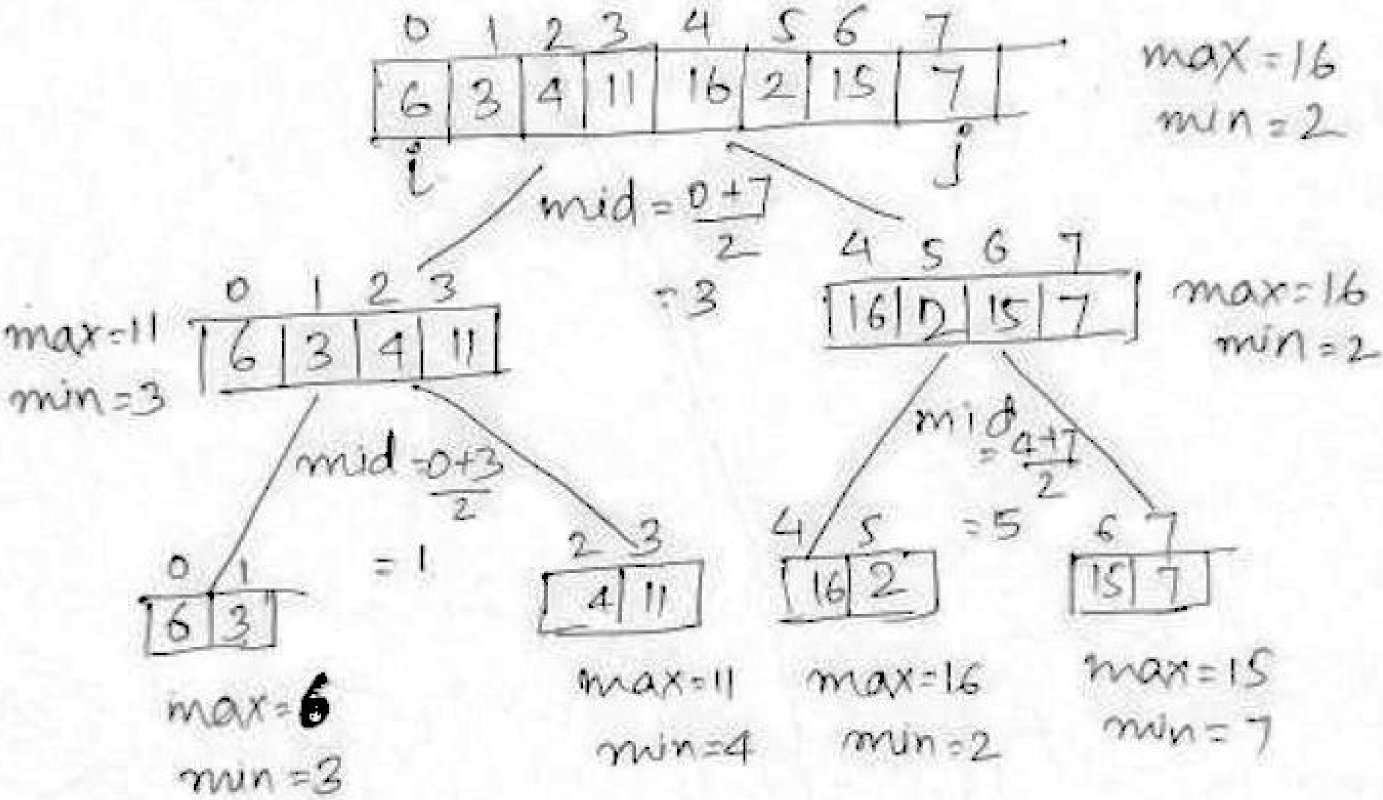
$T(n) = O(n)$

A	6	3	4	11	16	2	15	7	8
	0	1	2	3	4	5	6	7	8
	i	i	i	i	i	i	i	i	

max = $\cancel{6}$ $\cancel{11}$ 16

min = $\cancel{6}$ $\cancel{3}$ 2

Divide and Conquer Approach



$mid = \frac{i+j}{2}$ Small: $\boxed{n=2}$

```
DAC Max_min( A, i, j, max, min)
{   int mid
    if (i = j)
        max = min = A[i] ;
    elseif (i == j-1)
    { if (A[i] < A[j])
          max = A[j]; min = A[j];
      else
          max = A[i], min = A[j];
    }else{
          mid = (i+j)/2 ;

    DAC max_min( A, i, mid, max, min)
    DAC max_min ( A, mid+1 , max, min)


        if (max1 < max2)
            max = max2 ;
        else  max = max1 ;
        if ( min1 < min2 )
            min = min1 ;
        else
            min = min2 ;
```

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

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

$$= 2\left[2T\left(n/2^2\right) + 2\right] + 2$$

$$= 2^2 T\left(n/2^2\right) + 2^2 + 2$$

$$= 2^2\left[2T\left(n/2^3\right) + 2\right] + 2^2 + 2$$

$$= 2^3 T\left(n/2^3\right) + 2^3 + 2^2 + 2$$

$$\vdots$$

$$= 2^k T\left(n/2^k\right) + 2^k + 2^{k-1} + \cdots + 2^2 + 2$$

$$= 2^k * 1 + \left(2^k + 2^{k-1} + \cdots 2^2 + 2^1\right)$$

$$= 2^k + \frac{2\left(2^k - 1\right)}{2 - 1}$$

$$\boxed{\frac{n}{2^k} = 2 \Rightarrow \frac{n}{2} = 2^k \qquad 2^{k+1} = n}$$

$$= 2^k + 2^{k+1} - 2$$

$$= \frac{n}{2} + n - 2$$

$$= \frac{3n}{2} - 2 = \boxed{1.5n - 2} = O(n)$$