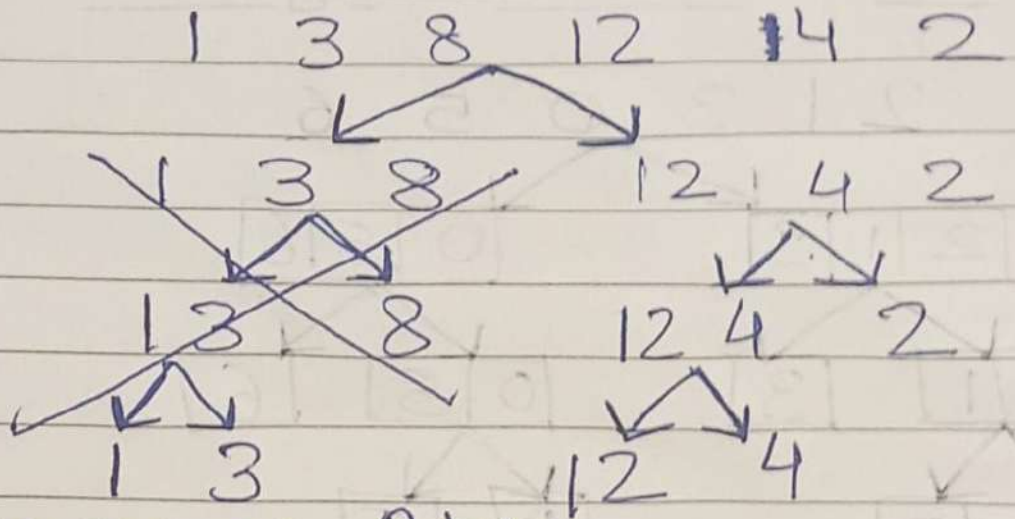
* Max_element in Bitonic Array



Recurrence Relation:

$$T(n) = T(n/2) + O(1)$$

$$a = 1,\ b = 2,\ d = 0$$

$$a = b^d \Rightarrow O(\log n)$$

Algorithm:

```
if (arr[mid]>arr[mid-1]&& arr[mid]>[mid+1])
   return [mid]
if(arr[mid]>arr[mid-1]&&arr[mid]<[mid+1])
   return max_element(arr, m+1, high)
else
   return max_element(arr, low, mid-1)
```

- Basically we compare the highest of left and least of right and discard one of them to maintain log(n).