Heap Sort

① Create the array

② Input values

③ Heapify

④ Swap the root with last ele

⑤ Again goto step 3

1	2	3	4	5	6	7	8
15	12	18	19	23	6	20	8

```
15  12
15  12  18
18  12  15  19
18  19  15  12
19  18  15  12  23
19  23  15  12  18
23  19  15  12  18  6
23  19  15  12  18  6  20
23  19  20  12  18  6  15
```

```
23  19  20  12  18  6  15  8
-----------------------------
1   2   3   4   5   6  7
8   19  20  12  18  6  15  [23]
19  8   20
20  8   19  12
20  12  19  8   18
20  18  19  8   12  6
20  18  19  8   12  6  15
15  18  19  8   12  6  [20] [23]
```

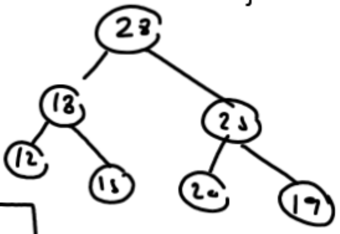
```
int main()
{
    int arr[8];
    int i,n;
    for(i=1;i<=7;i++){
        printf("Enter no:");
        scanf("%d",&n);
      ✓ insert(arr,i,n);
    }        7  19
    printf("The heap is:");
    for(i=1;i<=7;i++)
        printf("\n%d",arr[i]);
    for(i=7;i>=1;i--){
        arr[i]=del(arr,i);
    }
    printf("\nSorted array is :");
    for(i=1;i<=7;i++)
        printf("\n%d",arr[i]);
}
```

20
12
25
15
18
28
19

$n(\log_2 n)$

$n(\log_2 n)$

```
void insert(int arr[],int child,int n){
int par;                                    7    19
while(child>1){
    par=child/2;       3
    if(arr[par]>n){
        arr[child]=n;
        return;
    }
    arr[child]=arr[par];
    child=par;
}
arr[1]=n;
}
```



0	1	2	3	4	5	6	7
[illegible]	28	18	25	12	15	20	19

```
int del(int arr[],int n){
    int temp;              6
int x=arr[1];                5
arr[1]=arr[n];
n--;
int i=1,j=2*i;
while(j<=n){
    if(j<n){
    if(arr[j+1]>arr[j])
        j=j+1;
    }
    if(arr[i]<arr[j]){
        temp=arr[i];
        arr[i]=arr[j];
        arr[j]=temp;
        i=j;
        j=2*j;
    }
    else
        break;

}
return x;
}
```

1	2	3	4	5	6	7
20	18	19	12	15	25	28
i	j	j i			j	

$\boxed{25}$
x

COMPARITIVE TABLE OF SORTING ALGO

Algo	Best Case	Avg Case	Worst Case	Adaptive	Space Comp
① Bubble Sort	$O(n)$	$O(n^2)$	$O(n^2)$	Yes	$O(1)$
② Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	No	$O(1)$
③ Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$	Yes	$O(1)$
④ Merge Sort	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$	No	$O(n)$
⑤ QS	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	Yes	$O(n)$ or $O(\log n)$
⑥ HS	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	No	$O(1)$