

## Insertion Sort

### Design 3

Start check from  $A[k]$ , compare with  $A[k-1]$   
either  $A[k]$  right place, Go next  
 $A[k]$  wrong place,  $A[k-1]$  moved up.  
 $A[k]$  compare with  $A[k-2]$ .

insert\_sort (Stats, \*A, n) {

for i in range from 1 to n

  j = i

  temp = A[i]

  while j > 0 and cmp(temp < A[j-1])

    A[j] = A[j-1]

    j -= 1

  A[j] = temp

## Shell sort

Separates into Gaps, reduces until 1.

```
void shell_sort (stats, *A, n) {
```

$$\text{big\_K} = \frac{\log(3+2!)}{\log(3)}$$

```
for (to act as generator) {
```

Translate python code here.

```
}
```

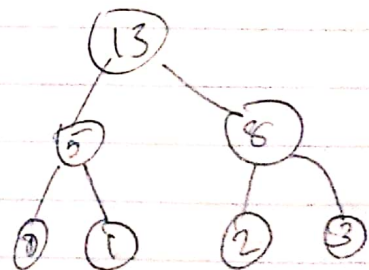
## Heap Sort

Max\_Child function ( )

fix\_heap function ( )

build\_heap function ( )

heap\_sort ( )



Max heap? parent node must be bigger  
or = than child



## Quick Sort

Divides into 2 sub arrays. by  
choosing a pivot.

elements  $<$  pivot

left of pivot

elements  $>$  pivot

right of pivot.

Partition()

Translate code from python to C