```
for j = 1 to A.length - 1 do
   key = A[j]
   // Insert A[j] into the sorted sequence A[0..j−1]
   i = j - 1
   while i >= 0 and A[i] > key do
      A[i + 1] = A[i]
      i = i - 1
   end while
   A[i + 1] = key
```

```
def print_pairs(arr):
   n = len(arr)
   for i in range(n):
      for j in range(i + 1, n):
         print(f"Pair: ({arr[i]}, {arr[j]})")
```

time	cost
c_1	1 +
c_2	1 +
c_3	n +
c_4	n - 1 +
c_5	1 +

$1 + 1 + n(n-1) \cdot 1$

$2 + \boxed{n^2} - n$

```
def power(x, n):
   result = 1
   base = x
   while n > 0:
      # If n is odd, multiply the result by the base
      if n % 2 == 1:
         result *= base
      # Square the base and halve the exponent
      base *= base
      n //= 2
   return result
```

$x^n = x^{n/2} \cdot x^{n/2}$

if x = even

or

x = odd

$x^n = x \cdot x^{n-1/2}$

If $n = 13$ bits = $\boxed{1101}$

total no of iter of while loop is the no. of times we want to halve the i/p, which is no of bits in the bin rep of your i/p

iter1 = 13 1101

iter2 = 6 110

iter 3 = 3 11

iter 4 ≈ 0

$\log_2 13$ = 4 steps to finish the loop