0	1	2	3	4	5	6	7	8	9
-8	-2	0	1	2	3	4	5	9	11
j					i				

2	0	1	3	9	7	-8	4	11	5

0	1	2	3	4	5	6	7	8	9
-8	0	1	2	3	4	5	7	9	11
ep	j								

$n, n?$

$= n^2$

$(n, n-1, n-2, n-3, n-4, n-5, n-6, n-7, n-8 \ldots\ldots)$

$(1, 2, 3, 4, 5, 6, \ldots\ldots n-4, n-3, n-2, n-1, n)$

$\frac{(n)(n+1)}{2} \approx n^2$

BS → TC?

BS → SC?

BS ↓ optimize sort

```python
def insertionSort(arr):
    l = len(arr)

    for i in range(1,l):
        j = i - 1
        while j >= 0 and arr[j] > arr[j + 1]:
            arr[j], arr[j + 1] = arr[j + 1], arr[j]
            j -= 1
```

-8	-2	0	1	2	3,4	5	7	10

j i

2	4	6	8	10	11

2	3	3	9

2	2	3	3	4	6	8	9	10	11

$T(N) = (\text{work done in pre+post area}) * H + (\text{No. of calls})^H$

$= (N+1)\log(N) + 2^{\log N}$

$= \underline{(N \log N + N)}$

$T(N) = 1 + 1 + T(N/2) + T(N/2) + N$

$T(N) = N + \cancel{2T(N/2)}$

$\cancel{2T(N/2)} = 2\frac{N}{2} + \cancel{2^2 T(N/4)}$

$\cancel{2^2 T(N/4)} = 2^2 \frac{N}{4} + \cancel{2^3 T(N/8)}$

$\cancel{2^3 T(N/8)} = 2^3 \frac{N}{8} + \cancel{2^4 T(N/16)}$

$\vdots$

$\cancel{T(1)} = 2^k \frac{N}{2^k} + 2^{k} T(1)$

$N, \frac{N}{2}, \frac{N}{4}, \frac{N}{8}, \frac{N}{16}, \ldots, 8, 4, 2, 1$

$T(N) = \sum_{0}^{k} N + 2^{kh}$

$\boxed{T(N) = KN + 2^{Kh}}$

$T(N) = N\log(N) + 2^{\log_2(N)}$

$\boxed{T(N) = N\log(N) + N}$

$T(N)$

```python
def mergeSort(arr, si, ei):
    if si == ei:
        return [arr[si]]

    mid = (si + ei) // 2
    leftHalf = mergeSort(arr, si , mid)
    rightHalf = mergeSort(arr, mid + 1 , ei)

    return mergeTwoSortedList(leftHalf, rightHalf)
```