

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

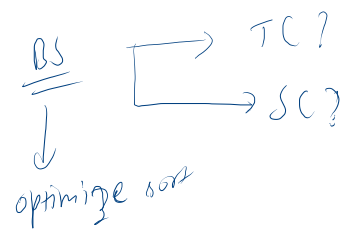
ep  $j$

$$h, h^2 = h^2$$

$$(h, h-1, h-2, h-3, h-4, h-5, h-6, h-7, h-8, \dots)$$

$$(1, 2, 3, 4, 5, 6, \dots, h-4, h-3, h-2, h-1, h)$$

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



```
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	2	3	3	7	6	8	9	10	11
---	---	---	---	---	---	---	---	----	----

2	7	6	8	10	11
---	---	---	---	----	----

2	3	3	9
---	---	---	---

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

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

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

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

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

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

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

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

$$(N \frac{N}{2}, 1 \frac{N}{4}, 1 \frac{N}{8}, 1 \frac{N}{16}, \dots, 1, 1, 1, 1)$$

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

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

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

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

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

```
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)
```

