2nd

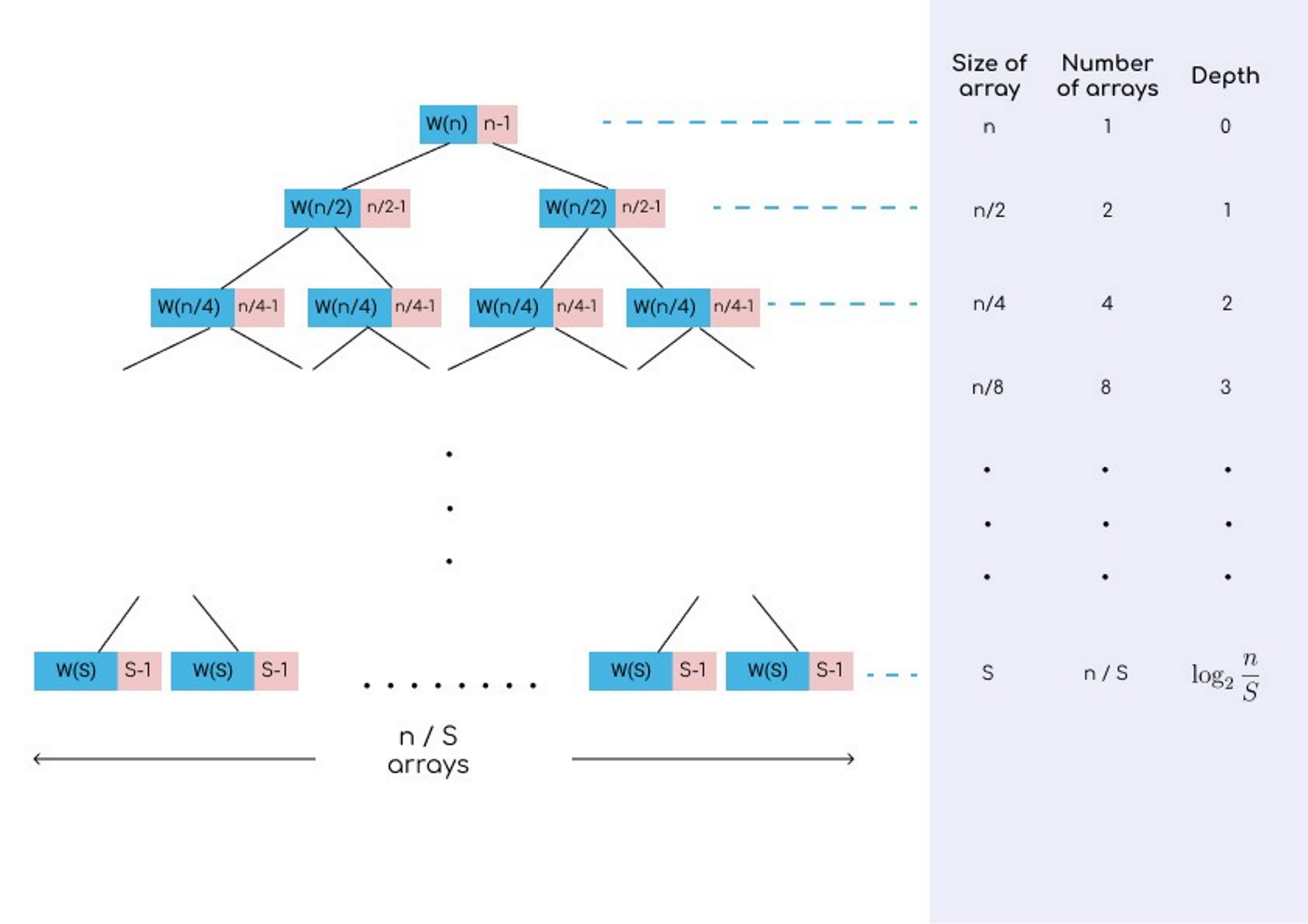
Tuesday, 2 February 2021 2:49 pm

```
def hybridSort(arr, start, end, S   ):
    mid = (start + end) // 2

    if end - start + 1 ≤ S :
        insertionSort (arr, start, end)
        return

    if end - start ≥ 2 :
        hybridSort (arr, start, mid, S)
        hybridSort (arr, mid+1, end, S)

    merge (arr, start, end)
```



Size of array	Number of arrays	Depth
n	1	0
n/2	2	1
n/4	4	2
n/8	8	3
.	.	.
.	.	.
.	.	.
S	n / S	$\log_2 \frac{n}{S}$

↳ For the layers up till 2nd last layer
(depth = 0 to $\log_2 \frac{n}{S} - 1$),

$$\# \text{ comparisons} = (n-1) + (n-2) + (n-4) + (n-8) + \ldots + (n - [k-1])$$

$$= n(\log_2 \tfrac{n}{S}) - (1 + 2 + 4 + 8 + \ldots + (k-1))$$

$$= n \log_2 \frac{n}{S} - \frac{2^{\log_2 \frac{n}{S}} - 1}{2-1}$$

$$= n \log_2 \frac{n}{S} - \frac{n}{S} + 1$$

↳ At the last layer (depth = $\log_2 \frac{n}{S}$),

# comparisons for each leaf $= \dfrac{(S-1)(S+2)}{4}$, assume average case for insertion sort.

$+ \ (S-1)$, for merge step.

∴ Total \# comparisons for last layer

$$= \left[\frac{(S-1)(S+2)}{4} + S - 1 \right] \times \frac{n}{S}$$

$$= \left[\frac{S^2 + S - 2 + 4S - 4}{4} \right] \times \frac{n}{S}$$

$$= \frac{nS^2 + 5nS - 6n}{4S}$$

↳ Total key comparisons

$$= n\log_2\frac{n}{s} - \frac{n}{s} + 1 + \frac{ns^2+5ns-6n}{4s}$$

$$= n\log_2\frac{n}{s} + \underbrace{\frac{5}{4}n - \frac{5}{2}\cdot\frac{n}{s} + \frac{1}{4}ns}_{O(ns)}$$

$$= O\left(ns + n\log\frac{n}{s}\right) \quad //$$