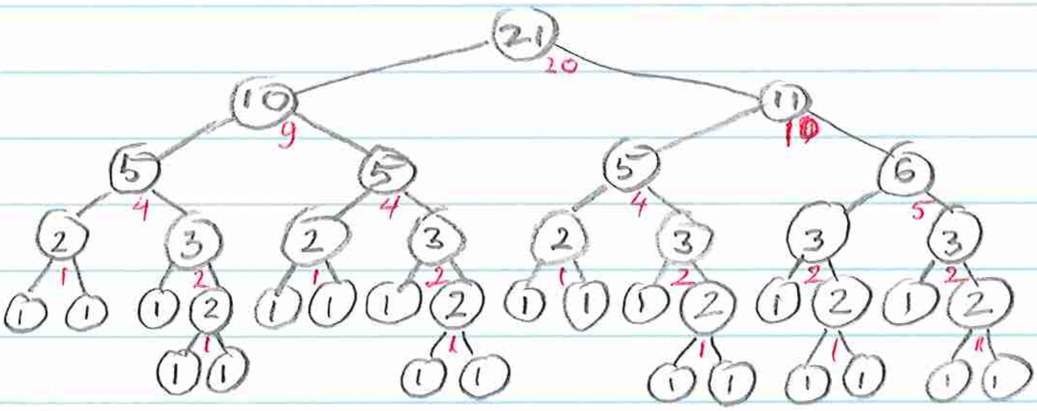
Worst-case number of comparisons done by Mergesort for $n = 21$ element array.



$$20 = 21 - 1 = 21 - 2^0$$
$$19 = 21 - 2 = 21 - 2^1$$
$$17 = 21 - 4 = 21 - 2^2$$
$$13 = 21 - 8 = 21 - 2^3$$
$$5 = 21 - 16 = 21 - 2^4$$

$= 74$

$$= 5 \cdot 21 - \sum_{i=0}^{4} 2^i$$
$$= 5 \cdot 21 - (2^5 - 1)$$
$$= 21 \cdot 5 - 2^5 + 1$$

The formula for the number of comps is:

$$n\lceil \lg(n+1) \rceil - 2^{\lceil \lg(n+1) \rceil} + 1$$

$n = 21$ $\quad n+1 = 22 \quad \lg 22 \approx 4.46 \quad \lceil \lg 22 \rceil = 5$

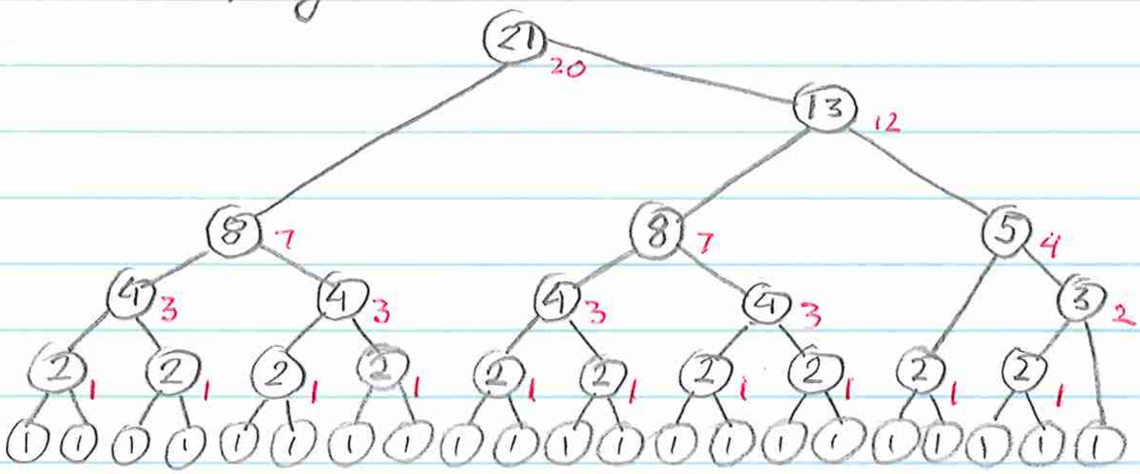
So, the number of comps must have been:

$$21 \cdot 5 - 2^5 + 1 = 74$$

That was <u>exactly</u> what we got!

I can't beat it!

Here is my attempt - I know it's optimal for linear merging.



= 20

= 12

= 18

= 14

= 10

74

the same number of comps as for Mergesort.

So, Mergesort's merges were worst-case optimal for $n = 21$ and linear merging.

Merge insertion sort (not discussed in class) would accomplish this in 66 comparisons only.

No sorting by comparison of keys can beat that: $\lceil \lg 21! \rceil = 66$.