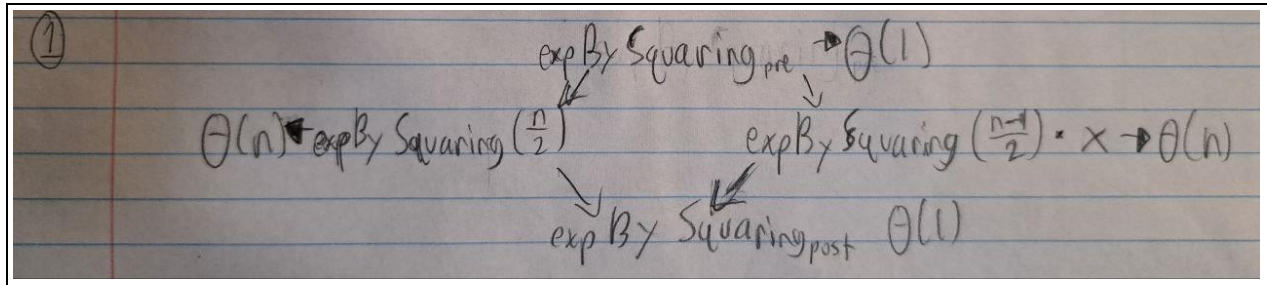


1

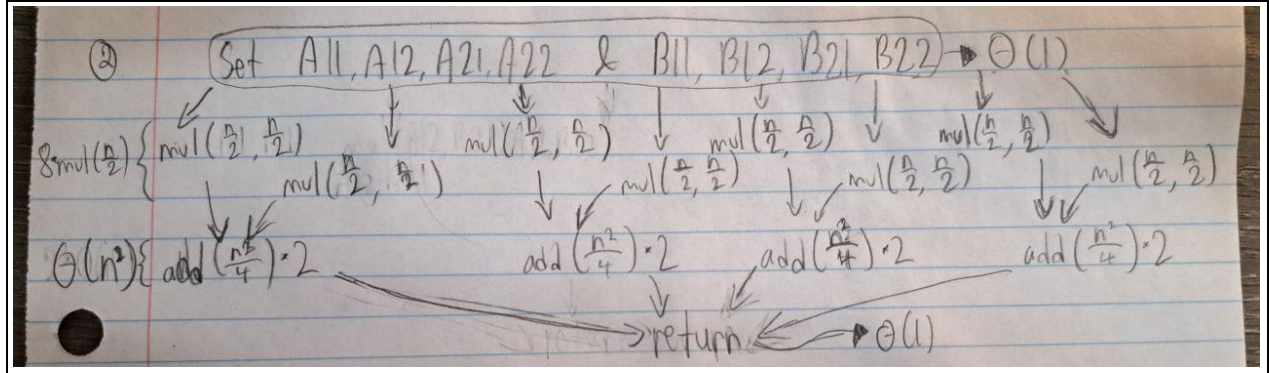
- Complexity: $\Theta(N) = \boxed{\Theta(\log(n))}$
- Dependencies:



- width = \boxed{n}
- $\sum p_i = \boxed{\Theta(n \cdot \log(n))}$
- $T_\infty = \boxed{\Theta(\log(n))}$, length = $\boxed{\log(n)}$

2

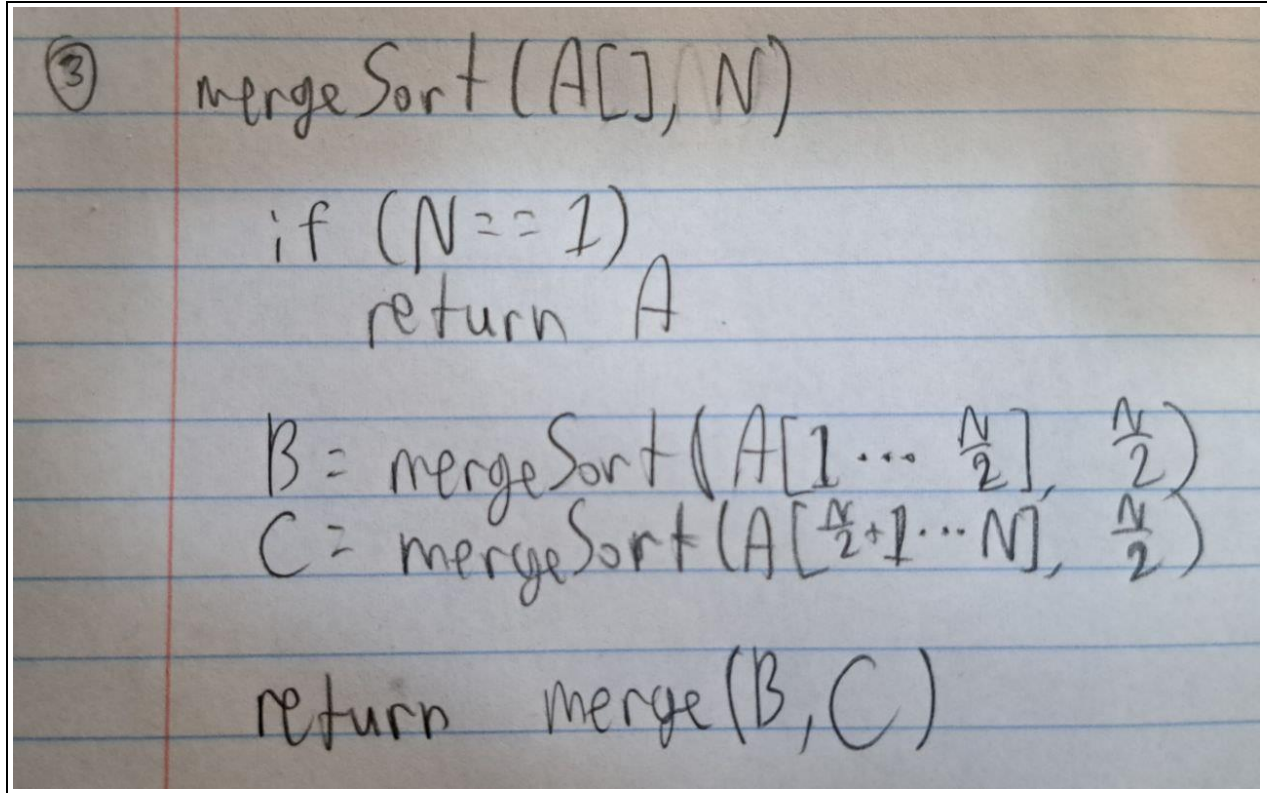
- **Complexity:** $\Theta(N) = 8 \cdot \text{mul}\left(\frac{n}{2}\right) + \Theta(n^2) \rightarrow \text{Master Theorem} \rightarrow a = 8, b = 2, f(n) = \Theta(n^2), \Theta(n^2) = O(n^{\log_2 8 - \epsilon})$ where $\epsilon > 0$ & $\epsilon \leq 1 \rightarrow T(n) = \Theta(n^{\log_2 8}) = \boxed{\Theta(n^3)}$
- **Dependencies:**



- **width** = $\boxed{8}$
- $\sum p_i = \boxed{8 \cdot \text{mul}\left(\frac{n}{2}\right) + 4 \cdot \Theta(n^2)}$
- $T_\infty = \boxed{\text{mul}\left(\frac{n}{2}\right) + \Theta(n^2)}, \text{length} = \boxed{\log(n)}$

3

- Algorithm:

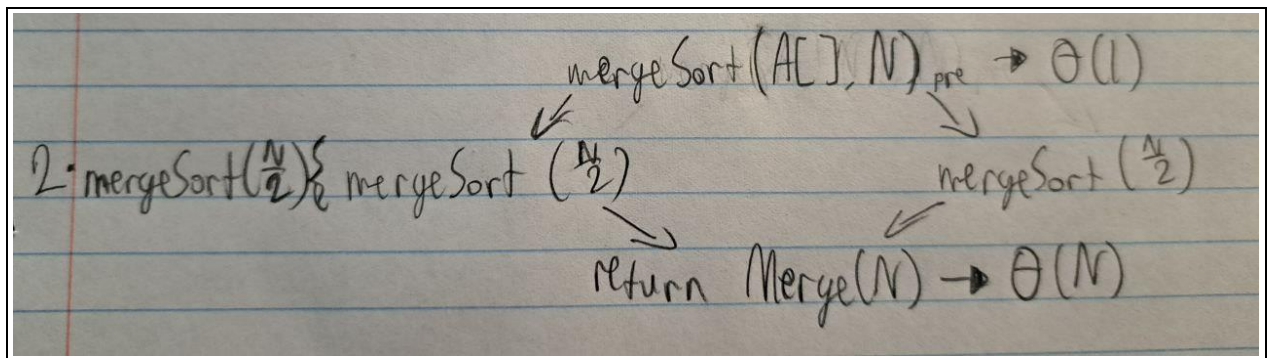


```

③ mergeSort(A[], N)
    if (N == 1)
        return A
    B = mergeSort(A[1... N/2], N/2)
    C = mergeSort(A[N/2+1... N], N/2)
    return merge(B, C)
  
```

- Complexity:** $\Theta(N) = \text{mrgSrt}\left(\frac{n}{2}\right) + \Theta(n) \rightarrow \text{Master Theorem} \rightarrow a = 1, b = 2, f(n) = \Theta(n), \Theta(n) = \Omega(n^{\log_2 1 + \epsilon})$ where $\epsilon > 0$ & $\epsilon \leq 1, 1 \cdot \Theta\left(\frac{n}{2}\right) \leq c \cdot \Theta(n)$ where $c \geq 0$ and $c < 1$ and n is sufficiently large, $\frac{3}{3} \rightarrow e, T(n) = \Theta(\Theta(n))$

- Dependencies:



```

graph TD
    A["mergeSort(A[], N)"] --> B["mergeSort(N/2)"]
    A --> C["mergeSort(N/2)"]
    B --> D["return Merge(N)"]
    C --> D
    A --> E["Θ(1)"]
    D --> F["Θ(N)"]
  
```

- The merge sort calls have the same processing time because they are recursive calls. The elemental atomic functions all have the same work.

- $width = \left\lceil \frac{n}{2} \right\rceil$
- $\sum p_i = 2mrgSrt\left(\frac{n}{2}\right) + \Theta(n) \Rightarrow \text{Master Theorem} \rightarrow a = 2, b = 2, f(n) = \Theta(n), \Theta(n) = \Theta(n^{\log_2 2}) \rightarrow T(n) = n^{\log_2 2} \log(n) = \boxed{\Theta(n \cdot \log(n))}$
- $T_\infty = mrgSrt\left(\frac{n}{2}\right) + \Theta(n) = \boxed{\Theta(\Theta(n))}, length = \boxed{2\log(n)}$

	Task Graph																																							
Processors	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28												
π_1	mergeSort(N/2)														(N/2)mergeSort[2]																									
π_2	mergeSort(N/4)												mergeSort(N/8)																		mergeSort(N/16)								→	
π_3																																								
π_4																																								