```java
public static ListNode mergeSort(ListNode head) {
    if (head == null || head.next == null)
        return head;

    ListNode mid = midNode(head);
    ListNode nHead = mid.next;
    mid.next = null;

    return mergeTwoLists(mergeSort(head), mergeSort(nHead));
}
```

$T(n)$

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

$2T(n/2) = 2 \cdot 2n/2 + 2^2 T(n/4)$

$2^2 T(n/4) = 2^2 \cdot 2n/4 + 2^3 T(n/8)$

$2^3 T(n/8) = 2^3 \cdot 2n/8 + 2^4 T(n/16)$

$\vdots$

$2^k T(1) = 2^k \cdot 2n/2^k + 2^{k+1} T(0)$

$T(n) = 2n\lambda + 2 \cdot 2^\lambda$

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

$= 2n \log_2(n) + 2n$

$T(n) = N \log N$

$n \quad n/2 \quad n/4 \quad \dots$

$a_0 = n$

$a_\lambda = 1$

$a_\lambda = a_0 r^{\lambda - 1}$

$1 = n \left(\frac{1}{2}\right)^\lambda$

$2^\lambda = n$

$\log_2 2^\lambda = \log_2(n)$

$\lambda = \log_2(n)$