1 Running Time

When we execute the function splitSwap(a, 0, n) its take an ammount of time c to execute the if condition, then the function recall itself two time and work on array of lenght $\frac{n}{2}$. Finally the function call anothe function named swapList, that take a time $O(n)$ to be executed. We call $T(n)$ time function.

$$\begin{aligned} T(n) &= c + 2T\left(\frac{n}{2}\right) + b\frac{n}{2} = \\ &= 3c + 4T\left(\frac{n}{4}\right) + bn = \\ &= 7c + 8T\left(\frac{n}{8}\right) + \frac{3}{2}bn = \\ &= (2^i - 1)c + 2^i T\left(\frac{n}{2^i}\right) + \frac{i}{2}bn \end{aligned} \tag{1}$$

The function stop to recall itself when $\frac{n}{2^i} = 1 \rightarrow i = \log_2(n)$. replacing it in the time equation, we get an running time $O\big(n\log(n)\big)$.

2 Mechanism of the Algorithm

the first time we call the function splitSwap(a, 0, n) the given matrix is cut in half and the position of the two halves is exchanged. then the function recall itself two time and repeat the described process to heach halves. At the end the array is inverted.
We can perform a faster algorithm using a for-loop.
if A[ ] is an array of lenght n

for i in range $\frac{n}{2}$:

tmb = A[i]
A[i] = A[n - i]
A[n - i] = tmb