

Generating Functions and Linear Recurrences

Kieran Rimmer *

September 2020

Merge sort is a divide and conquer algorithm wherein a data structure to be sorted is recursively split in two until sorting is trivial, and then the sorted substructures are merged until the full structure is sorted.

For simplicity, let us consider only data which is sized in powers of two. Then the worst case running time

$$T_n$$

for a naive implementation of merge sort can be approximated with:

$$\begin{aligned} T_n &= c_1 + T_{(2^n/2)} + T_{(2^n/2)} + c_2 \cdot 2^n \\ &= c_1 + 2 \cdot T_{(2^n/2)} + c_2 \cdot 2^n \end{aligned} \quad (1)$$

This relation yields a sequence, which has a corresponding *generating function*:

$$\langle T_0, T_1, T_2, T_3 \dots \rangle \leftrightarrow F(x) = T_0 + T_1 \cdot x + T_2 \cdot x^2 + T_3 \cdot x^3 \dots \quad (2)$$

The sequence can be broken down into a sum of sequences, for which convenient closed form summations exist *if convergence is assumed*:

$$\begin{aligned} &\langle c_1, c_1, c_1, c_1 \dots \rangle \leftrightarrow c_1/(1-x) \\ &\langle c_2, 2 \cdot c_2, 4 \cdot c_2 \dots 2^n \cdot c_2 \rangle \leftrightarrow c_2/(1-2x) \\ &+ \langle 0, 2 \cdot T_0, 2 \cdot T_1 \dots 2 \cdot T_{n-1} \rangle \leftrightarrow 2xF(x) \end{aligned}$$

$$\langle T_0, T_1, T_2, T_3 \dots \rangle \leftrightarrow F(x) = c_1/(1-x) + c_2/(1-2x) + 2xF(x) \quad (3)$$

*just some Jimbo