7) We ~~are~~ consider the radix-2 FFT algorithm, then the algorithm actually divides the whole series of DFT into halves containing even and odd Terms and we work our way back.

This algorithm of radix 2 works only for step size = 2^m $(m \in \mathbb{N})$, and we work out the complexity for them only.

say $\tilde{w}_q$ represent the DFT for the frequency point q.

$$\tilde{w}_q = \frac{1}{\sqrt{n}} \sum_{p=0}^{2^m-1} \exp\left(-i \frac{2\pi q p}{n}\right) w_p .$$

where $\{w_p\}$ are data points.

we can divide the series as:

$$\frac{1}{\sqrt{n}} \sum_{p=0}^{n/2-1} w_{2p} \left(\exp\left(-\frac{i 2\pi q \cdot p}{n/2}\right)\right).$$

$$+ \sum_{p=0}^{n/2-1} w_{2p+1} \exp\left(-\frac{i 2\pi q}{n}\right) \exp\left(\frac{i 2\pi q p}{n/2}\right).$$

we can write This as:

$$\frac{1}{\sqrt{n}} \Big\{ DFT \left(n/2 \text{ points (even points)} \right) \Big)$$

$$+ \phi^q \, DFT \left(n/2 \text{ points (odd points)} \right) \Big).$$

where $\phi = \exp\left(\frac{-i 2\pi}{n}\right)$.

Again we can divide the $n/2$ points into $n/4$ points each

so the chain goes as:

$$DFT_n = DFT^e_{n/2} + \phi^q \, DFT^o_{n/2}$$

$$= DFT^{ee}_{N/4} + DFT^{eo}_{N/4} \cdot (\phi)^{cs} + \left(DFT^{oe}_{N/4} + DFT^{oo}_{N/4}\right)(\phi)^{cs}$$

~~..~~ i.e we keep multiplying phases and computing DFTs.

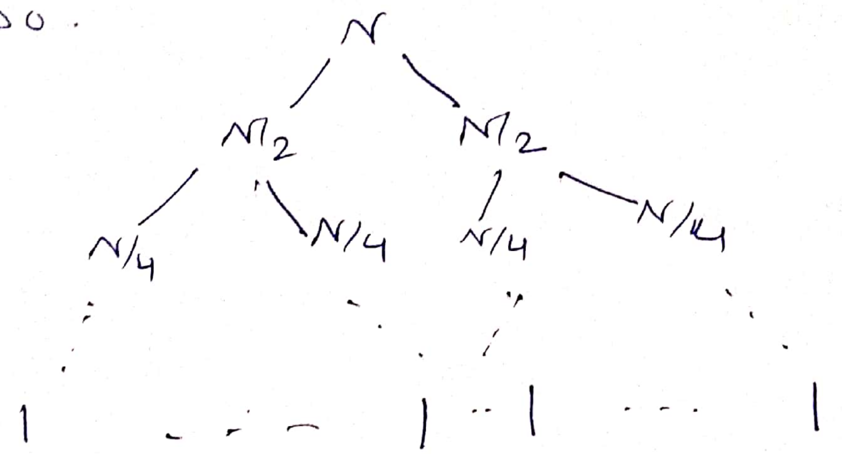
How long can this split go on?

Until we reach only single numbers.

But DFT of single number is that number itself. What truely remains is multiplying the phase factor and adding them up.

What we need to do is do this recursively multiply phases and add up.

so.



At each step of triangle what ~~mat..~~ matters are the number of operations we're doing at each step.

so at each step we have $O(N)$ operational complexity and.

There are $m = \log_2 N$ steps to compute them at. So the net complexity of the algorithm is given by:

$$O(N \log_2 N).$$ (By product of complexities).