Algorithms & Datastructures.
HW#3
Problem 3.2

b) let $A = (a_n a_{n-1} \dots a_1)_2$ assuming n to be a power of 2
$B = (b_n b_{n-1} \dots b_1)_2$

A divide & conquer algorithm can be implemented by breaking A & B into 2 integers of $n/2$ bits each.

$$A = \underbrace{(a_n \dots a_{n/2+1})_2}_{A_1} \cdot 2^{n/2} + \underbrace{(a_{n/2} \dots a_1)_2}_{A_2}$$

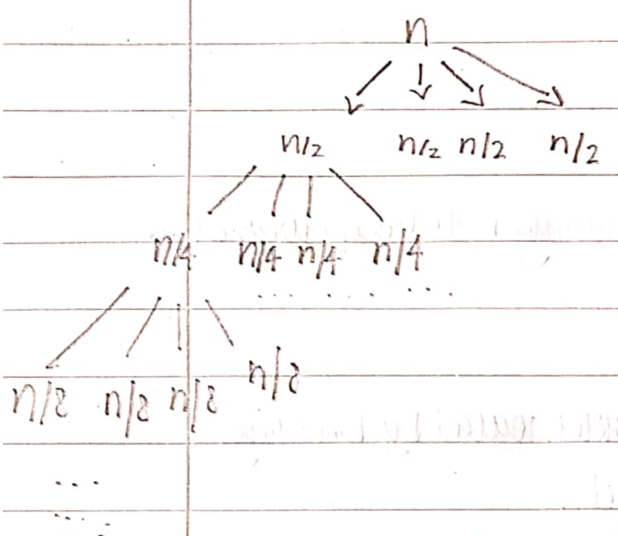
$$B = \underbrace{(b_n \dots b_{n/2+1})_2}_{B_1} \cdot 2^{n/2} + \underbrace{(b_{n/2} \dots b_1)_2}_{B_2}$$

$$(AB) = A_1 \cdot B_1 \cdot 2^n + (A_1 \cdot B_2 + A_2 \cdot B_1) \cdot 2^{n/2} + A_2 \cdot B_2$$

$\Rightarrow$ reduced to 4 multiplication of $n/2$ bit integers, 3 additions of integers with $2n$ bits and 2 shifts.

c) $T(n) = 4T(n/2) + cn$

d). $T(n) = 4T(n/2) + cn.$



n

$2(n)$

$4(n)$

$8(n)$

$T(1)$ $= n + 2(n) + 4(n) + \cdots 4^{\log n}(T)(1)$

e) Master Theorem.

$n^\wedge \log_b a = n^2$

$T(n) = \Theta(n^2).$

$\therefore$ proven

$$= n \sum_{i=0}^{\log_2 n - 1} 2^i + \Theta(4^{\log n})$$

$$= n(2^{\log n} - 1)/(2-1) + \Theta(2^2)^{\log n}$$

$$= \Theta(n^2) + \Theta(n^2) = \Theta(n^2).$$