GF(2^4) 상에서의 Karatsuba Multiplication

https://youtu.be/Ska3_6jP2jw

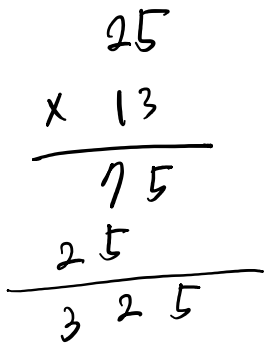
Karatsuba Multiplication

- 1960년도에 발견된 큰 수에 대해 효과적인 곱셈 알고리즘

- 일반적으로 n 자리 수 두개를 곱하기 위해서 한 자리 수의 곱셈을 n^2 번 해야함

  ex) 25 x 13 연산을 하기 위해 한 자리 수의 곱셈 4번이 필요
  Karatsuba 연산 사용 시, 한 자리 곱셈 횟수가 최대 $2^{\log_2 3}$ 개로 줄어듦.

1) 3 x 5
2) 3 x 2
3) 1 x 5
4) 1 x 2



https://ko.wikipedia.org/wiki/%EC%B9%B4%EB%9D%BC%EC%B6%94%EB%B0%94_%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98

Karatsuba Multiplication

- Karatsuba 알고리즘 기본 단계
  - 큰 두 수인 x 와 y의 곱을 자릿수가 x ,y의 절반인 수들의 곱셈 3번, 덧셈, 시프트 연산 이용

    x와 y를 k진법의 n자리 수라고 가정, n보다 작은 양수인 m에 대하여 x, y를 쪼갬

$$x = x_1 k^m + x_0$$
$$y = y_1 k^m + y_0$$

$$z_0 = x_0 y_0$$
$$z_1 = x_1 y_0 + x_0 y_1$$
$$z_2 = x_1 y_1$$

z_0, z_1, z_2를 구성하기 위해 총 4번의 곱셈이 사용 됨.

$$xy = (x_1 k^m + x_0)(y_1 k^m + y_0)$$
$$= k^{2m} x_1 y_1 + k^m (x_1 y_0 + x_0 y_1) + x_0 y_0$$
$$= k^{2m} \cdot z_2 + k^m z_1 + z_0$$

Karatsuba Multiplication

- Karatsuba : 덧셈을 여러 번 연산하여 xy를 3번의 곱셈을 통해 연산 방법

$$z_0 = x_0 y_0$$
$$z_2 = x_1 y_1$$

$$z_1 = (x_1 y_1 + x_1 y_0 + x_0 y_1 + x_0 y_0) - x_1 y_1 - x_0 y_0 = x_1 y_0 + x_0 y_1$$

$$z_1 = (x_1 + x_0)(y_1 + y_0) - z_2 - z_0$$

Karatsuba Multiplication

- $k = 10$일 때, 12345 x 6789 연산 방법

$$12345 = 12 * 10^3 + 345$$

$$6789 = 6 * 10^3 + 789$$

$$z_2 = 12 * 6 = 72$$

$$z_0 = 345 * 789 = 272205$$

$$z_1 = (12 + 345) * (6 + 789) - z_2 - z_0 = 357 * 795 - 72 - 272205 = 283815 - 72 - 272205 = 11538$$

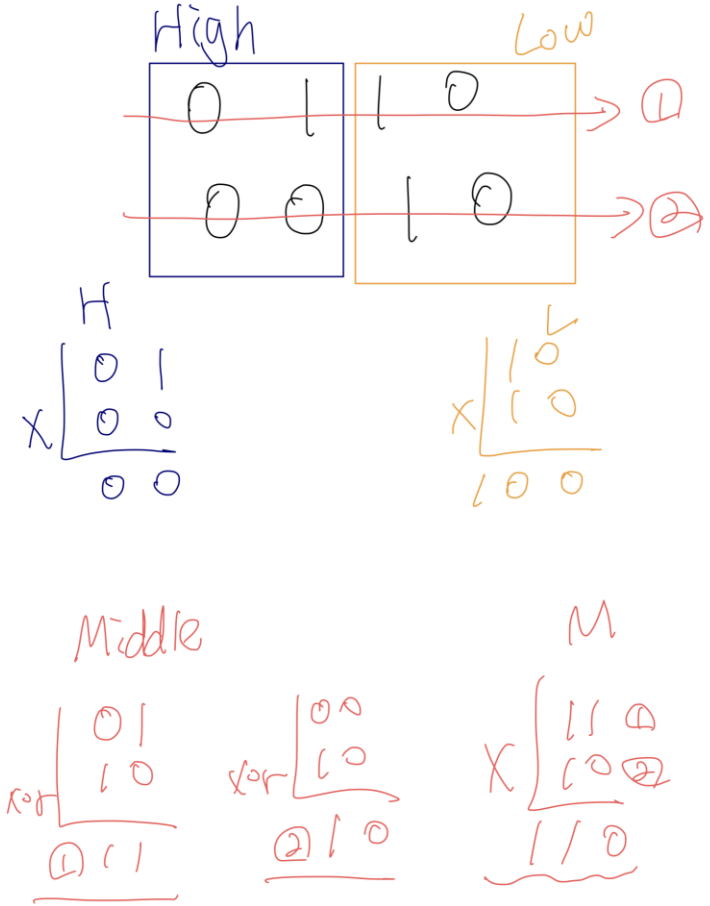
$$z_2 * (10^3)^2 + z_1 * 10^3 + z_0 = 72 * (10^3)^2 + 11538 * 10^3 + 272205 = 83810205$$

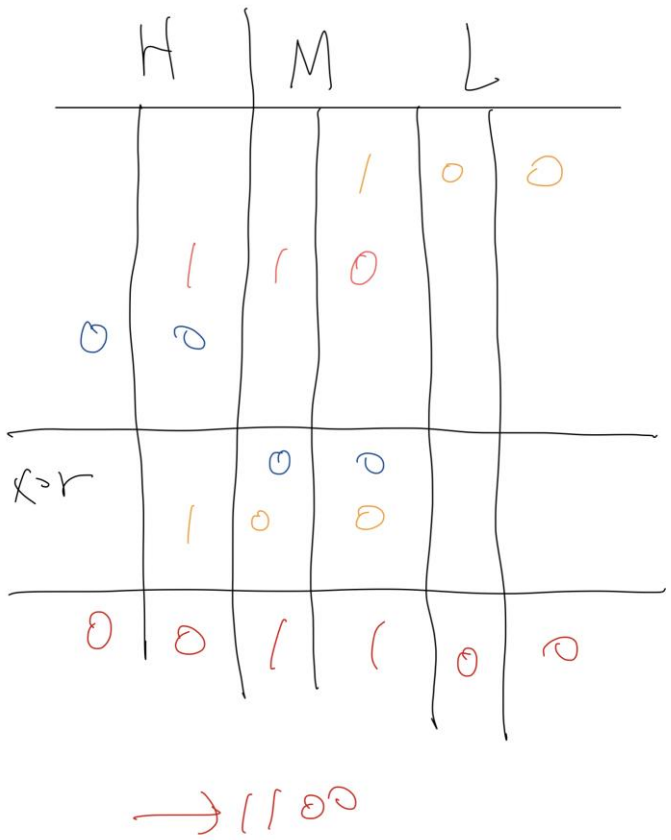
Karatsuba Multiplication

- 효율성 분석
  - 기본 단계는 모든 k와 m에 대해 동작
    - M이 n/2일 때, 가장 효율적
    - 양의 정수 q에 대해 n = 2^q & 재귀가 n = 1일 때, 멈춘다면, 한 자리 곱셈의 횟수= 3^q
  - 충분히 큰 n에 대해서는 기존 곱셈보다 적은 횟수의 시프트 연산과 한 자리 곱셈 수행
    - 작은 n에 대해서는 추가적인 덧셈과 시프트 연산으로 인해 속도가 기존보다 느려짐
    - 컴퓨터 플랫폼 마다 기준이 되는 경계가 다름
      - 보통 $2^{230} \approx 2^* 10^{96}$ 일때, 카라추바 알고리즘이 더 빠른 연산

GF(2^4) 상에서의 Karatsuba Multiplication

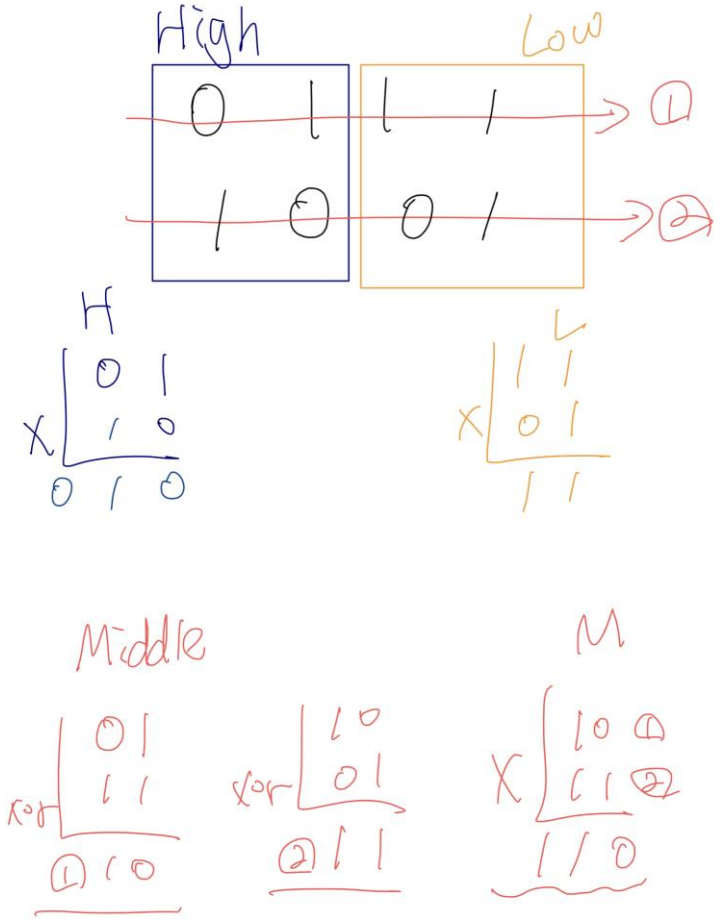
- a = 0110, b = 0010 일 때, 0110 * 0010 연산 방법

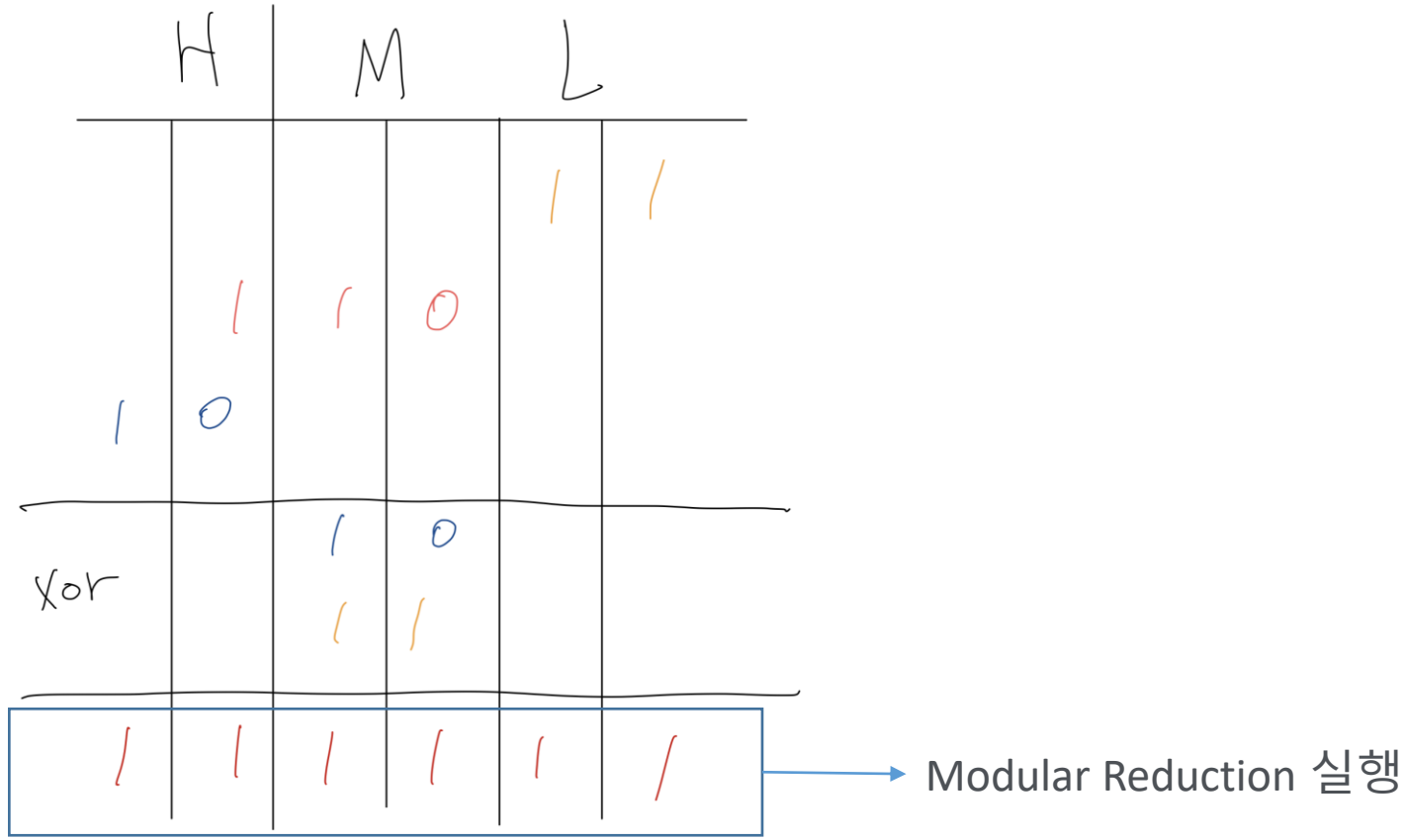




GF(2^4) 상에서의 Karatsuba Multiplication

- a = 0111, b = 1001 일 때, 0111 * 1001 연산 방법





Q & A