Polynomial Multiplication

TOTAL POINTS 3

1. For $n = 1024$, compute how many operations will the faster divide and conquer algorithm from the lectures perform, using the formula $3^{\log_2 n}$ for the number of operations. **1 / 1 point**

- [ ] 1024
- [ ] 1048576
- [x] 59049

✓ Correct
$\log_2 n = \log_2 1024 = 10$, so $3^{\log_2 n} = 3^{10} = 59049$.

2. What is the key formula used in the faster divide and conquer algorithm to decrease the number of multiplications needed from 4 to 3? **1 / 1 point**

- [x] $a_1b_0 + a_0b_1 = (a_0 + a_1)(b_0 + b_1) - a_0b_0 - a_1b_1$
- [ ] $a_1(b_0 + b_1) = a_1b_0 + a_1b_1$
- [ ] $a_0 + b_0 = a_1 + b_1$
- [ ] $(a_0 + a_1)(b_0 + b_1) = a_0b_0 + a_0b_1 + a_1b_0 + a_1b_1$

✓ Correct
Correct! This means that we only need to do 3 multiplications a_0b_0, a_1b_1 and $(a_0 + a_1)(b_0 + b_1)$ instead of 4 multiplications a_0b_0, a_1b_1, a_0b_1 and a_1b_0.

3. *(This is an advanced question.)* **1 / 1 point**

How to apply fast polynomial multiplication algorithm to multiply very big integer numbers (containing hundreds of thousands of digits) faster?

- [x] For a number $A = \overline{a_1a_2\ldots a_n}$ with n digits create a corresponding polynomial $a(x) = a_1x^{n-1} + a_2x^{n-2} + \cdots + a_n$. Then $a(10) = A$. Do the same with number $B = \overline{b_1b_2\ldots b_n}$ and create polynomial $b(x)$. Multiply polynomials $a(x)$ and $b(x)$, get polynomial $c(x) = \overline{c_1c_2\ldots c_n}$. If we create a number $C = \overline{c_1c_2\ldots c_n}$, it is almost the same as product of A and B, but some of its "digits" may be 10 or bigger. If the last "digit" is 52, for example, make the last digit just 2 and add 5 to the previous digit. Go on until all the digits are from 0 to 9.

  Suppose we need to multiply numbers 13 and 24. The correct result is 312. To get this result, we first create polynomials $a(x) = x + 3$ and $b(x) = 2x + 4$ corresponding to numbers 13 and 24 respectively. We then use Karatsuba's algorithm to multiply those polynomials and get polynomial $c(x) = 2x^2 + 10x + 12$. To get the answer, we need to compute $c(10) = 2 \times 10^2 + 10 \times 10 + 12$. You see that some of the coefficients of polynomial c are not digits, because they are bigger than 9. To fix that, for each such coefficient from right to left we subtract 10 from it and add 1 to the previous coefficient: $c(10) = 2 \times 10^2 + 10 \times 10 + 12 = 2 \times 10^2 + 11 \times 10 + 2 = 3 \times 10^2 + 1 \times 10 + 2 = 312$.

- [ ] For number A, create a polynomial $a(x) = A$, for number B create a polynomial $b(x) = B$, multiply those polynomials and get the answer.

  Suppose we need to multiply numbers 13 and 24. The correct result is 312. To get this result, we first create polynomials $a(x) = 13$ and $b(x) = 24$ corresponding to numbers 13 and 24 respectively. We then use Karatsuba's algorithm to multiply those polynomials and get polynomial $c(x) = 312$. Now we know that $13 \times 24 = 312$.

✓ Correct
First we need to convert number with n digits to polynomial with n coefficients in $O(n)$ time. Then we need to multiply two polynomials of degree n in $O(3^{\log_2 n})$ time. After that, we need to convert the polynomial back to number and "fix" it in $O(n)$. The total time for multiplication of the numbers will be $O(n) + O(3^{\log_2 n}) + O(n) = O(3^{\log_2 n})$ as opposed to $O(n^2)$ time for the grade school number multiplication algorithm.