

به نام خدا

تمرین اول درس الگوریتم های پیشرفته

امین محمدلو 401422169

1: بله با استراتژی تقسیم و غلبه می توان با کم تر کردن عملیات ضرب روش بهینه تری را برای ضرب دو عدد ارائه داد.

شبه کد الگوریتم (این الگوریتم با نام karatsuba شناخته می شود):

```
Procedure
karatsuba(num1,
num2)
    if (num1 < 10) or (num2 < 10)
        return num1*num2
    /* calculates the size of the numbers */
    m = max(size_base10(num1), size_base10(num2))
    m2 = m/2
    /* split the digit sequences about the middle */
    high1, low1 = split_at(num1, m2)
    high2, low2 = split_at(num2, m2)
    /* 3 calls made to numbers approximately half the size */
    z0 = karatsuba(low1,low2)
    z1 = karatsuba((low1+high1),(low2+high2))
    z2 = karatsuba(high1,high2)
    return (z2*10^(2*m2))+((z1-z2-z0)*10^(m2))+z0
```

در این الگوریتم در ابتدا اگر اعداد کوچکتر از 10 باشند ضرب را به صورت عادی محاسبه میکنیم (حالت پایه در روش بازگشتی).

سپس عددی که تعداد ارقام بیشتری دارد را مشخص میکنیم تا عدد  $n/2$  که در ویدیو کمکی توضیح داده شد را پیدا کنیم.

سپس اعداد را با توجه به استراتژی divide and conquer به دو قسمت تقسیم میکنیم و زیرمسائل به وجود آمده را دوباره با همین روش محاسبه میکنیم و در انتها تجميع میکنیم.

مرحله تقسیم با کامنت `/* split the digit sequences about the middle */`

مرحله غلبه با کامنت `/* 3 calls made to numbers approximately half the size */`

و مرحله تجميع در خط آخر شبه کد مشخص شده است.

2: کوچکترین زیر مسئله همان حالت پایه یعنی  $a < 10$  or  $b < 10$  است.

4: چون در این الگوریتم به جای چهار عمل ضرب ، سه عمل ضرب داریم (در ویدیو کمکی نیز توضیح داده شد) پس پیچیدگی زمانی این الگوریتم برابر است با  $T(n) = 3T(n/2) + O(n)$

در نتیجه  $O(n^{1.59})$