

تمرین اول درس الگوریتم پیشرفته

آرمان اکبرزاده

401422014

در این تمرین قصد داریم ضرب دو عدد بزرگ را به روش تقسیم و غلبه انجام دهیم ، از آن جهت که خواسته های سوال در بخش های مختلف کمی در هم تنیده هستند ، سعی میکنیم به طور کلی و دقیق شبهه کد را توضیح دهیم تا ابهامات برطرف شود.

ابتدا نیاز به ذکر است که الگوریتم تقسیم و غلبه برای ضرب دو عدد ، کاراتسوبا نام دارد و حال سعی میکنیم که شبهه کد این الگوریتم را توضیح دهیم :

```
Function karatsuba(X, Y){  
    // base case  
    If (X < 10 and Y < 10)  
        return X*Y;  
  
    // determine the size of numbers  
    size = maximum(Length(string(X)), Length(string(Y)))  
  
    // Split X and Y  
    n = ceil(size/2)  
    p = power(10,n) // equivalent to 10^n  
    a = floor(X/p)  
    b = X%p  
    c = floor(Y/p)  
    d = Y%p  
  
    // Recur until base case  
    ac = karatsuba(a,c)  
    bd = karatsuba(b,d)  
    e = karatsuba(a+c, b+d) - ac - bd  
  
    // Return the Multiplication  
    return power(10,2*n)*ac + power(10,n)*e + b  
}
```

مرحله اول تعیین حداکثر اندازه (تعداد ارقام) اعداد ورودی است که با تبدیل اعداد به رشته ها و محاسبه طول رشته ها مطابق شکل به دست می آید.

مرحله بعدی این است که X را به a و b و Y را به c و d تقسیم کنیم تا قسمت های دوم آنها یعنی b و d همیشه رقم اضافی را در صورت فرد بودن اندازه (n) بدست آورند، برای مثال اگر X برابر 2023 باشد و Y برابر 3003 باشد ، داریم :

$$n = 2, p = 100, a = 20, b = 23, c = 30, d = 3$$

در نتیجه داریم :

$$Y = 102 * 30 + 3 \text{ و } X = 102 * 20 + 23$$

اکنون مراحل بالا را برای ضرب بیشتر اعداد تقسیم شده در مراحل گفته شده تکرار میکنیم تا

اینکه حالت پایه بدست آید یعنی وقتی X و Y کمتر از 10 می شوند.

و آخرین مرحله این است که معادله حاصل را مطابق شکل برگردانیم.

مرحله تقسیم در شبه کد با کامنت Split مشخص شده است و غلبه با صدا زدن توابع بازگشتی

است و تجمیع نیز return آخر در شبه کد است.

کوچک ترین زیر مسئله نیز حالت پایه است که در شکل مشخص است.

در مورد پیچدگی زمانی نیز میتوان گفت که ما با کاهش تعداد ضرب از 4 به 3 ، میتوانیم به

الگوریتمی با زمان بهتر دست پیدا کنیم (در ویدیو کمکی نیز اشاره ای کلی شد) با استفاده از

درخت بازگشتی داریم : $T(n) = 3T(n/2) + O(n)$ که مرتبه زمانی این الگوریتم حدود n به

توان 1.59 میشود که به وضوح از n به توان 2 کمتر است.