۱- ایّا می توان با استراتژي تقسیم و غلبه روشی بهینه نسبت به روش بالا براي عملیات ضرب دو عدد صحیح ارایه داد؟

بله ، ایده الگور یتم به ای صورت اس تکه هر عدد را به ۲بخش تق سیم میک نیم ،به طور مثال دو عدد داریم به m,n و طول هر عددرا هم با k نشان می دهیم, عد د اول به صورت m=a*10^k/2+b نشان داد پس :

مرحله divide and conquer

M = 4563

45*100+63   K=4

نشان میدهیم عدد بعدی را هم به ه مین ترتیب

= 6842 24*10^4/2 + 86

n = c*10^k/2 + d

از ضرب m*n داریم :

(45*10^2+63)*(24*10^2+86) =(1080)*10^4 +(ad+bc)*10^2 + 149

برای حل (ad+bc) میتوا نیم بنوی سیم)تجمیع(

(ac+bc+ad+bd) - ac -bd =(ad+bc)=(a+b)*(c+d)-ac-bd

و با این کار الگورتم به ینه شده چون ۴تا ضرب به دوتا ضرب تب دیل شد ه .
شبه کد :

pre dir=ltr>

```
Algorithm Multiply(a, b):
// Conquer
if |a| == 1 or |b| == 1 then
return A * B
else
// Split
n ← max(|A|, |B|)
xL, xR ← Split(A, n)
yL, yR ← Split(B, n)

// Combine
z0 ← multiply(xL, yL)
z1 ← multiply(xL + xR, yR + yL)
z2 ← multiply(xR, yR)
return z2 * 10^(2*m) + (z1 - z2 - z0) * 10^m + z0
end
```

</pre>

2-

در این الگوریتم به جای ۴ تا subproblems از ۳تا استفاده میکنیم:

Three subproblems:

$a= X 1 y1$ $d = X 2 y2$ $e = (X 1+X2)(y1+y2)-a-d$

4-

الگوریتم سه ضرب و چند جمع را انجام می دهد و روی اعداد کوچکتر که تقریباً نصف اندازه اصلی هستند جابجا می شود.

time complexity بهاین صورت میشود:

Then $xy= ar^n +e r^n/2+d$

$T(n)= 3T (n/2)+O(n)$

$T(n) = O(n ^log3) = O (n^1.584)$