Homework #1

(Due: March 21)

GROUP NUMBER: 18

Group Members		
Name	SBU ID	% Contribution
Venkata Ravi Teja Takkella	113219890	33.3
Manish Reddy Vadala	114190006	33.3
Ranjith Reddy Bommidi	114241300	33.3

Collaborating Groups

Group Number	Specifics (e.g., specific group member? specific task/subtask?)
NA	NA

External Resources Used

	Specifics (e.g., cite papers, webpages, etc. and mention why each was used)
1.	NA
2.	
3.	
4.	
5.	

(1) Given the known recurrence that computes,

$$\binom{n}{r} = \begin{cases} 0 & \text{if } r<0 \text{ or } r>n \\ 1 & \text{if } n=r=0. \\ \binom{n-1}{r-1} + \binom{n-1}{r} & \text{otherwise} \end{cases}$$

Now a generalised algorithm is given as.

$$S(n,r) = \begin{cases} 0 & \text{if } r<0 \text{ or } r>n \\ 1 & \text{if } n=r=0 \\ a \cdot S(n-1,r-1) + b \cdot S(n-1,r) & \text{otherwise} \end{cases}$$

Now we need to design an algorithm that computes $S(n,r)$ for all $r \in [0,n]$ in $O(n \log n)$

Now for simplicity of the algorithm, Let us take $n = 2^k$ & we can generalise this algorithm for $\forall n$ at the end.

Now let's try to generalize this algorithm.

For $n = 1$

$$S(1,0) = \underbrace{a\,S(0,-1)}_{\text{since } r<0} + \underbrace{b\,S(0,0)}_{n=r=0} = b$$

$a \cdot 0$ $\quad$ b

$$S(1,1) = \underbrace{a\,S(0,0)}_{n=r=0} + \underbrace{b\,S(0,1)}_{r>n} = a.$$
$$a \qquad 0$$

for $n=1$ $\quad S(1,0) = b \quad S(1,1) = a.$

For $n=2$

$$S(2,0) = \underbrace{a\,S(1,-1)}_{r<0} + \underbrace{b\,S(1,0)}_{S(1,0)=b} = b^2$$
$$0 \qquad b^2$$

$$S(2,1) = a\,S(1,0) + b\,S(1,1) \quad \begin{bmatrix} S(1,0)=b \\ S(1,1)=a \end{bmatrix}$$
$$= ab + ba$$
$$= 2ab$$
$$S(2,2) = a\,S(1,1) + \underbrace{b\,S(1,2)}_{r>n \Rightarrow 0}$$
$$= a^2$$

So for $n=2$ $\quad S(2,0) = b^2 \quad S(2,1) = 2ab \quad S(2,2) = a^2.$

Where as for $n=3$

$$S(3,0) = a\,S(2,-1) + b\,S(2,0)$$
$$= b^3$$
$$S(3,1) = a\,S(2,0) + b\,S(2,1)$$
$$= ab^2 + 2ab^2 = 3ab^2$$
$$S(3,2) = a\,S(2,1) + b\,S(2,2)$$
$$= 2a^2b + a^2b = 3a^2b$$
$$S(3,3) = a\,S(2,2) + b\,S(2,3)$$
$$= a^3.$$

So For $n=3$

$S(3,0) = b^3$, $S(3,1) = 3ab^2$ $S(3,2) = 3a^2b$ $S(3,3) = a^3$.

If we look at this sequence.

We can take a polynomial which looks like

$ax+b$. & it's powers.

For $n=1$

$\Rightarrow$ $ax+b$ $a = S(1,1)$ $b = S(1,0)$

For $n=2$

$\Rightarrow$ $(ax+b)^2 = a^2x^2 + 2abx + b^2$

$a^2 = S(2,2)$ $2ab = S(2,1)$ $b^2 = S(2,0)$

For $n=3$

$\Rightarrow$ $(ax+b)^3 = a^3x^2 + 3a^2bx^2 + 3ab^2x + b^3$

$a^3 = S(3,3)$, $3a^2b = S(3,2)$ $3ab^2 = S(3,1)$

$b^3 = S(3,0)$

So we can say that

$(ax+b)^n$ expansion will give the terms

which are $S(n,n)$, $S(n,n-1)$ . . . $S(n,0)$

That's what was asked to be found.

Now how to reach $(ax+b)^n$

To calculate for $n=2$

We basically do $(ax+b)(ax+b)$

This can be done $O(2\log 2)$ time using FFT.

FFT says that the multiplication of 2'nd degree takes $O(n\log n)$ time complexity.

Similarly.

$(ax+b)$ $n=1$ ↓ $O(2\log 2)$

$(ax+b)^2$ $n=2$

↓

$(ax+b)^2(ax+b)^2 = (ax+b)^4 = n=4$ ↓ $O(4\log 4)$

⋮

$(ax+b)^n =$ where $n = 2^K$.

$O(2^K \log 2^K)$

Since we took $n=2^K$, we jump at every power of 2.

So the total time complexity to reach $(ax+b)^n$

$= T =$ $O(2\log 2) + O(2^2 \log 2^2)$

$+ \cdots + O(2^K \log 2^K)$

$$T = \sum_{i=1}^{K} O(2^i \log 2^i) \rightarrow \sum_{i=1}^{K} O(2^i i).$$

So $T/2 = \sum_{i=1}^{K} O(2^{i-1} i)$

So $T - T/2 = T/2 = \sum_{i=1}^{K} O(2^i i) - \sum_{i=1}^{K} O(2^{i-1} i)$

So $T/2 = [O(2^1 \cdot 1) - O(2^0 \cdot 1)] + [O(2^2 \cdot 2) - O(2^1 \cdot 2)]$

$+ \cdots$

$+ [O(2^K \cdot K) - O(2^0 \cdot 1)]$

$T/2 = O(2^K \cdot K)$

$T = 2\,O(2^K \cdot K) \Rightarrow O(2^K \cdot K)$

upper bound is still the same

We know $n = 2^K \quad K = \log n$

$T = (n \log n)$.

So the time complexity $O(n \log n)$.

So we have proved that for $n = 2^K$

to find $(ax+b)^n$ which has the terms equivalent to

$S(n,n), S(n,n-1) \cdots S(n,2)$

$\Rightarrow S(n,r) \quad \forall r \in [0,n]$.

We can do this in $O(n \log n)$ time comp

If $n \neq 2^k$.

Let us take $2^K < n < 2^{K+1}$.

Since we have computed $2^K, 2^{K-1}, 2^{K-2}, \ldots, 1$ in $O(n \log n)$ time.

We can use the previous calculated powers to reach n between $2^K \to 2^{K+1}$ in at max K steps. In worst case complexity of

$$\underbrace{O(n \log n)}_{\text{To reach } 2^K} + \underbrace{\sum_{j \in (\text{powers required to reach } n)} O(2^{K-j} \log 2^{K-j})}_{\Downarrow \text{ worst case} \Rightarrow O(n \log n)}$$

$\Rightarrow O(n \log n)$.

So to reach any n, we take $O(n \log n)$ complexity.

(2) Special case to general case.

given recursive function.

$$H(n,r) = \begin{cases} \alpha & \text{if } r = 0 \\ \beta & \text{if } r = N+1 \\ \gamma_r & \text{if } n = 0 \text{ \& } 1 \le r \le N, \\ a\,H(n-1,r-1) + b\,H(n-1,r) + c\,H(n-1,r+1) & \text{otherwise} \end{cases}$$

where $a, b, c, \alpha, \beta, \gamma_1 \cdots \gamma_N \in \mathbb{R}$.

Also given that there exists an algorithm for $N \in \mathbb{N}$ & $N' \in [1, N]$ $H(N', r)$ for all $r \in [1, N]$ can be calculated in $O(N \log N)$ time provided $\alpha = \beta = 0$.

Let us call this algorithm as CBZ.

So let's try to first expand our $H(n,r)$

For $n = 0$

$H(0,0) = \alpha \quad r = 0.$

$H(0,1) = \gamma_1 \quad H(0,2) = \gamma_2 \quad \cdots \quad H(0,N) = \gamma_N$

$H(0, N+1) = \beta. \quad r = N+1$

Similarly let's try for $n = 1$

$H(1,0) = \alpha \quad r = 0$

$H(1,1) = aH(0,0) + bH(0,1) + cH(0,2)$

$= a\alpha + b\gamma_1 + c\gamma_2$

$$H(1,2) = aH(0,1) + bH(0,2) + cH(0,3)$$
$$= a\gamma_1 + b\gamma_2 + c\gamma_3$$

⋮

Let us now try to calculate for $n=2$.

$$H(2,0) = \alpha \quad \gamma=0$$

$$H(2,1) = aH(1,0) + bH(1,1) + cH(1,2)$$
$$= a\alpha + b(a\alpha + b\gamma_1 + c\gamma_2) + c(a\gamma_1 + b\gamma_2 + c\gamma_3)$$
$$= a\alpha + ab\alpha + (b^2+ac)\gamma_1 + 2bc\gamma_2 + c^2\gamma_3.$$

⋮

Now what we'll do is, to find similar equations for CBZ where $\alpha = \beta = 0$.

So there

If $n=0$

$H(0,0) = 0 \quad \gamma=0$

$H(0,1) = \gamma_1$, $H(0,2) = \gamma_2$ ... $H(0,N) = \gamma_N$

$H(0,N+1) = 0 \quad \gamma = N+1$

$H(1,0) = 0 \quad \gamma=0$

$$H(1,1) = aH(0,0) + bH(0,1) + cH(0,2)$$
$$= b\gamma_1 + c\gamma_2$$
$$H(1,2) = a\gamma_1 + b\gamma_2 + c\gamma_3.$$

Similarly for $n=2$

$H(2,0) = 0$

$H(2,1) = aH(1,0) + bH(1,1) + cH(1,2)$

$= b(b\gamma_1 + c\gamma_2) + c(a\gamma_1 + b\gamma_2 + c\gamma_3)$

$= (b^2 + ac)\gamma_1 + 2bc\,\gamma_2 + c^2\gamma_3$.

⋮

So what we observe is basically the α & β terms are missing over here.

One Algorithm:

Find the additional α & β terms & add them up to each cell of the $CBZ(n,r)$ $\forall r \in [0,N]$ vector to get $H(n,r)$.

Now to find α & β terms. What we can do is replace all γ's with 0's in our $H(n,r)$ algorithm So we can just get terms of α & β.

The α & β terms follows a sequence over here.

Let us try to show that in a matrix form.

of α & βs.

We've taken all γs to be 0 initially.

$n=0$ α 0 0 - - - — -- - - - β

$n=1$ α $a\alpha$ 0 - - — - - - - .. $c\beta$ β

$n=2$ α $a(1+b)\alpha$ $a^2\alpha$ - - — — - $c^2\beta$ $(1+b)c\beta\beta$. —①

$n=3$ α -- - — — — — -- β.

⋮

$n=n'$ α - - — — — — - - β. Let us take * $n' = 2^k$

So these are the terms which we need to add for our vector $CBZ(n', r)$ to get $H(n', r)$.

Now to reach the $n = n'$th line, we no need to calculate for each n before.

What we try to do over here is, let's replace. $n=1$ line as $n=0$ & send it over to CBZ. But for CBZ we need $r=0$ & $r \geq N+1$ as 0 since $\alpha=\beta=0$.

So the new r vector being sent to CBZ looks like

$a\alpha$ 0 - - - - - $c\beta$

So if we try to find for $n=2$ in $O(N \log N)$ using ↳ which is CBZ.

It gives out. $n=1$ in new form.

0 0 $a\alpha$ 0 . - - . 0 $c\beta$ 0

1 0 $ab\alpha$ $a^2\alpha$ - - . $c^2\beta$ $bc\beta$ 0. — ②

So in the new generated vector

$\Rightarrow$ 0 $ab\alpha$ $a^2\alpha$ - - - - $c^2\beta$ $cb\beta$ 0.

We have missed few terms for $r=1$ & $r=N$

which are same as of $n=0$ in (2).

So what we do now is to add the $n=0$ line to our generated vector. And it turns out to be

0 $a(1+b)\alpha$ $a^2\alpha$ - - - - $c^2\beta$ $c(1+b)\beta$ 0.

This is same as $n=2$ from (1). at start & end. with α, β missing

Now we take these set of terms as new γ's & send it to CBZ to get the $n=4$ which 2^{nd} n in new CBZ.

Let's try to calculate that too.

0 0 $a(1+b)\alpha$ $a^2\alpha$ - - - - $c^2\beta$ $c(1+b)\beta$ 0

1 0 $a(b+b^2+ac)\alpha$ $a^2(1+2b)\alpha$ - - - - $(1+2b)c^2\beta$ $c(b+b^2+ac)\beta$ 0

2 0 $a(b^2+b^3+ac+3ab)\alpha$ $a^2(2b+3b^2+2ac)\alpha$ - - - - - - - 0

So if we observe $n=2$ here which is our 4^{th} in it.

The terms missing from (1) are same as our γ's which are our $n=0$.

So we add them.

And the time taken to get $n=2$ is $O(N \log N)$ given by CBZ for any n.

So if we add our γ's again to $r=2$, we get.

$a(1+b+b^2+b^3 \quad a^2(1+2b+3b^2 \quad - - - -$
$+ac+3abc)\alpha \quad +2ac)\alpha$

which is our required line in (1)

with α & β missing at our start & end.

So basically we go from

$n=1 \rightarrow n=2 \rightarrow n=4 \rightarrow - - - \rightarrow n=n/2$.

we can also find $n=5$ directly from $n=2$ but the missing terms go till b^2 which are not present in $n=2$ to add. So this addition of missing terms is possible only for $n=1 \rightarrow 2 \rightarrow 4 \rightarrow \cdots n/2$.

So we go till $n/2$ & give that line as γ's to CBZ to get n in $O(N \log N)$ again.

So the recurrence equation looks like.

$$T(n) = T(n/2) + \underbrace{O(N \log N)}_{\text{to get } N/2 \text{ as } \gamma\text{'s from CBZ.}} + \underbrace{O(N)}_{\text{to add } n'=n/2 \text{ to the obtained vector.}} + \underbrace{O(1)}_{\text{To add } \alpha \& \beta \text{ start \& end}}$$

$$T(n) = T(n/2) + O(N \log N)$$

Basically the n in T( ) & N in O( ) are not same.

This recurrence steps go till height of $\log N$ & adding $O(N \log N)$ $\log N$ times since $n = N$ over here.

$\Rightarrow$ So total time complexity $= O(N \log N \times \log N)$

$= O(N \log^2 N)$.

This is to generate the vector only in α's & β's

We get the $CBE(N, r)$ in $O(N \log N)$ which gives us all the γ's.

So if we add these both vectors we get $H(N, r)$

$=$ Total calculation

$= O(N \log^2 N) + O(N \log N)$

$\simeq O(N \log^2 N)$

Hence this algorithm can prove that $H(N, r)$ for all $r \in [1, N]$ can be obtained in $O(N \log^2 N)$ time without imposing any constraints on α, β.

But what if $N \neq 2^k$. Then in such case, we take $\lceil N/2 \rceil$ for going to the half point & similarly recursively down.

So after $\lceil n/2 \rceil$, we can find the n^{th} element
& add the missing terms from $\lceil n/2 \rceil^{th}$ row.
This can be proved in the similar way as above
& it does $O(N \log^2 N)$ time too.

3.

a) writing down the algorithm we derived in slides 17-18

Algorithm:

Input: nxn matrices X & Y

output: $Z \leftarrow XY$

(1) If X & Y are 1×1 matrices then

$Z \leftarrow X \cdot Y$

(2) else

{Recursion :- - - - }

(3) [Given $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}_{nxn}$ $Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}_{nxn}$

$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}_{nxn}$]

$K_0 = X_{12}(Y_{11} + Y_{21})$

$K_1 = X_{21}(Y_{12} + Y_{22})$

$K_2 = (X_{21} - X_{12})(Y_{11} + Y_{22})$

$K_3 = (X_{11} - X_{12}) Y_{11}$

$$K_4 = (X_{11} - X_{21})(Y_{12} - Y_{11})$$

$$K_5 = (X_{22} - X_{12})(Y_{21} - Y_{22})$$

$$K_6 = (X_{22} - X_{21})\, Y_{22}$$

(4) {Compute Output Matrix} ---

$$Z_{11} = K_0 + K_3$$

$$Z_{21} = K_0 + K_2 + K_5 + K_6$$

$$Z_{12} = K_1 - K_2 + K_3 + K_4$$

$$Z_{22} = K_1 + K_6$$

The matrices multiplied are differe
from strassen i.e., basically $K_0, K_1 \ldots K_6$
compared to the $P_{11}, P_{22} \ldots P_{72}$

The total no. of sums & substractions
here are 10 for generating K's &
8 in generating final matrix
where as the algorithm over here
we brought the original matrix to the giv
5 types, where for the first three types
we need 3 multiplications & second
two types need 4 multiplications which is
7 in total.

0 in strassen, we didn't try to

where as ...
bring in this to this format of 5 types of matrices, we generated new matrices P_{11}, P_{22} using the original matrix & finally brought the multiplication by applying some operations on them.

b) To Prove the variant, we need to show

$$Z_{11} = X_{11}Y_{11} + X_{12}Y_{21}$$
$$Z_{12} = X_{11}Y_{12} + X_{12}Y_{22}$$
$$Z_{21} = X_{21}Y_{11} + X_{22}Y_{21}$$
$$Z_{22} = X_{21}Y_{12} + X_{22}Y_{22}$$

$$Z_{11} = P_0 + P_3 - P_4 + P_6$$
$$= X_0Y_0 + X_3Y_3 - X_4Y_4 + X_6Y_6$$
$$\Rightarrow (X_{12}+X_{21})(Y_{12}+Y_{21}) + X_{21}(Y_{11}-Y_{21})$$
$$- (X_{11}+X_{12})Y_{12} + (X_{11}-X_{21})(Y_{11}+Y_{12})$$
$$\Rightarrow X_{12}Y_{12} + X_{21}Y_{12} + X_{12}Y_{21} + X_{21}Y_{21} + X_{21}Y_{11} - X_{21}Y_{2}$$
$$- X_{11}Y_{12} - X_{12}Y_{12} + X_{11}Y_{11} - X_{21}Y_{11}$$
$$+ X_{11}Y_{12} - X_{21}Y_{12}$$

$\therefore Z_{11} \Rightarrow X_{11}Y_{11} + X_{12}Y_{21}$

$Z = P + P$

$z_{12} = r_2 + r_4$

$\Rightarrow x_2y_2 + x_4y_4$

$\Rightarrow x_{12}(y_{22} - y_{12}) + (x_{11} + x_{12})y_{12}$

$\Rightarrow x_{12}y_{22} - x_{12}y_{12} + x_{11}y_{12} + x_{12}y_{12}$

$\therefore z_{12} \Rightarrow x_{12}y_{22} + x_{11}y_{12}$

$z_{21} = p_1 + p_3$

$\Rightarrow x_1y_1 + x_3y_3$

$\Rightarrow (x_{21} + x_{22})y_{21} + x_{21}(y_{11} - y_{21})$

$\Rightarrow x_{21}y_{21} + x_{22}y_{21} + x_{21}y_{11} - x_{21}y_{2}$

$\therefore z_{21} \Rightarrow x_{22}y_{21} + x_{21}y_{11}$

$z_{22} = p_0 - p_1 + p_2 + p_5$

$\Rightarrow x_0y_0 - x_1y_1 + x_2y_2 + x_5y_5$

$\Rightarrow (x_{12} + x_{21})(y_{12} + y_{21}) - (x_{21} + x_{22})y_{21}$

$+ x_{12}(y_{22} - y_{12}) + (x_{22} - x_{12})(y_{21} + y_{2}$

$\Rightarrow x_{12}y_{12} + x_{12}y_{21} + x_{21}y_{12} + x_{21}y_{21}$

$- x_{21}y_{21} - x_{22}y_{21} + x_{12}y_{22} - x_{12}y_{12}$

$+ x_{22}y_{21} - x_{12}y_{21} + x_{22}y_{22} - x_{12}y_{22}$

$\therefore z_{22} = x_{21}y_{12} + x_{22}y_{22}$

Hence we can show that it correctly multiplies two $n \times n$ matrices where as if $n \neq 2^k$

we can extend $n \rightarrow$ next 2^k & keep 0's everywhere it is extendend & it gives the matrix multiplication in $7T(n/2) + \Theta(n^2)$ time complexity.

(c) Now we have to derive the algorithm in Figure 1 using the approach shown in slides 17-18 of Lecture 3.

Basically we need to bring the original matrix into a sum form such that these matrices multiplied with one y gives Z in 7 unique multiplications.

Given $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$ $Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$

$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$

Since we already mentioned how we'll proceed for $n \neq 2^k$. Let us go with $n = 2^k$.

Based on the approach in slides.

$$\underbrace{\begin{bmatrix} x_{11} & x_{12} & 0 & 0 \\ x_{21} & x_{22} & 0 & 0 \\ 0 & 0 & x_{11} & x_{12} \\ 0 & 0 & x_{21} & x_{22} \end{bmatrix}}_{X} \times \underbrace{\begin{bmatrix} y_{11} \\ y_{21} \\ y_{12} \\ y_{22} \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} z_{11} \\ z_{21} \\ z_{12} \\ z_{22} \end{bmatrix}}_{Z}$$

Now let's convert X to our 5-types discussed in class.

$$X_2 = \underbrace{\begin{bmatrix} x_{21} & -x_{21} & 0 & 0 \\ x_{21} & -x_{21} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\Delta_1 \text{ (1 Multiplication)}} + \underbrace{\begin{bmatrix} x_{11}-x_{21} & x_{12}+x_{21} & 0 & 0 \\ 0 & x_{22}+x_{21} & 0 & 0 \\ 0 & 0 & x_{11} & x_{12} \\ 0 & 0 & x_{21} & x_{22} \end{bmatrix}}_{X'}$$

$$X' = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -x_{12} & x_{12} \\ 0 & 0 & -x_{12} & x_{12} \end{bmatrix}}_{\Delta_2 \text{ (1 Multiplication)}} + \underbrace{\begin{bmatrix} x_{11}-x_{21} & x_{12}+x_{21} & 0 & 0 \\ 0 & x_{22}+x_{21} & 0 & 0 \\ 0 & 0 & x_{11}+x_{12} & 0 \\ 0 & 0 & x_{21}+x_{12} & x_{22}- \end{bmatrix}}_{X''}$$

$$X'' = \begin{bmatrix} 0 & x_{12}+x_{21} & x_{12}+x_{21} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} x_{11}-x_{21} & 0 & -x_{12}-x_{21} & 0 \\ 0 & x_{22}+x_{21} & 0 & 0 \\ 0 & 0 & x_{11}+x_{12} & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & X_{12}+X_{21} & X_{12}+X_{21} & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & -X_2-X_{21} & 0 & X_{22}-X_1 \end{bmatrix}$$

Δ_3 (1 Multiplication) $\qquad X'''$

$$X'' = \begin{bmatrix} X_{11}-X_{21} & 0 & (X_{11}-X_{21}) & 0 \\ & & -(X_{11}+X_{12}) & \\ 0 & 0 & 0 & 0 \\ 0 & 0 & X_{11}+X_{12} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & X_{22}+X_{21} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & X_{22}-X_{12} & 0 & X_{22}- \\ & -(X_{22}+X_{21}) & & \end{bmatrix}$$

Δ_4 (2 Multiplications) $\qquad \Delta_5$ (2 Multipli

So $XY = Z$

$= (\Delta_1+\Delta_2+\Delta_3+\Delta_4+\Delta_5)\cdot Y = Z$

This whole takes 7 unique multiplic

So using this approach Let's get the algorith

The unique multiplications found are.

(1) $(X_{12}+X_{21})(Y_{12}+Y_{21})$

(2) $X_{21}(Y_{11}-Y_{21})$

(3) $(X_{21}+X_{22})Y_{21}$

(4) $(X_{11}+X_{12})Y_{12}$

(5) $X_{12}(Y_{22}-Y_{12})$

(6) $(X_{11}-X_{21})(Y_{11}+Y_{12})$

(7) $(X_{22}-X_{12})(Y_{21}+Y_{22})$

So the algorithm can be written like :

Input : $n \times n$ matrices X & Y.

Output : $Z \leftarrow XY$

(1) If X & Y are 1×1 matrices then $Z \leftarrow XY$.

(2) else

{ Recursion ---- }

(3)

$$K_0 = (X_{12} + X_{21})(Y_{12} + Y_{21})$$

$$K_1 = X_{21}(Y_{11} - Y_{21})$$

$$K_2 = (X_{21} + X_{22})\, Y_{21}$$

$$K_3 = (X_{11} + X_{12})\, Y_{12}$$

$$K_4 = X_{12}(Y_{22} - Y_{12})$$

$$K_5 = (X_{11} - X_{21})(Y_{11} + Y_{12})$$

$$K_6 = (X_{22} - X_{12})(Y_{21} + Y_{22})$$

{ Generate Output Matrix } ----

(4)

$$Z_{11} = K_0 + K_1 - K_3 + K_5$$

$$Z_{12} = K_3 + K_4$$

$$Z_{21} = K_1 + K_2$$

$$Z_{22} = K_0 - K_2 + K_4 + K_6$$

Hence this is the algorithm of Figure 1.

Eq using the approach shown in slides.