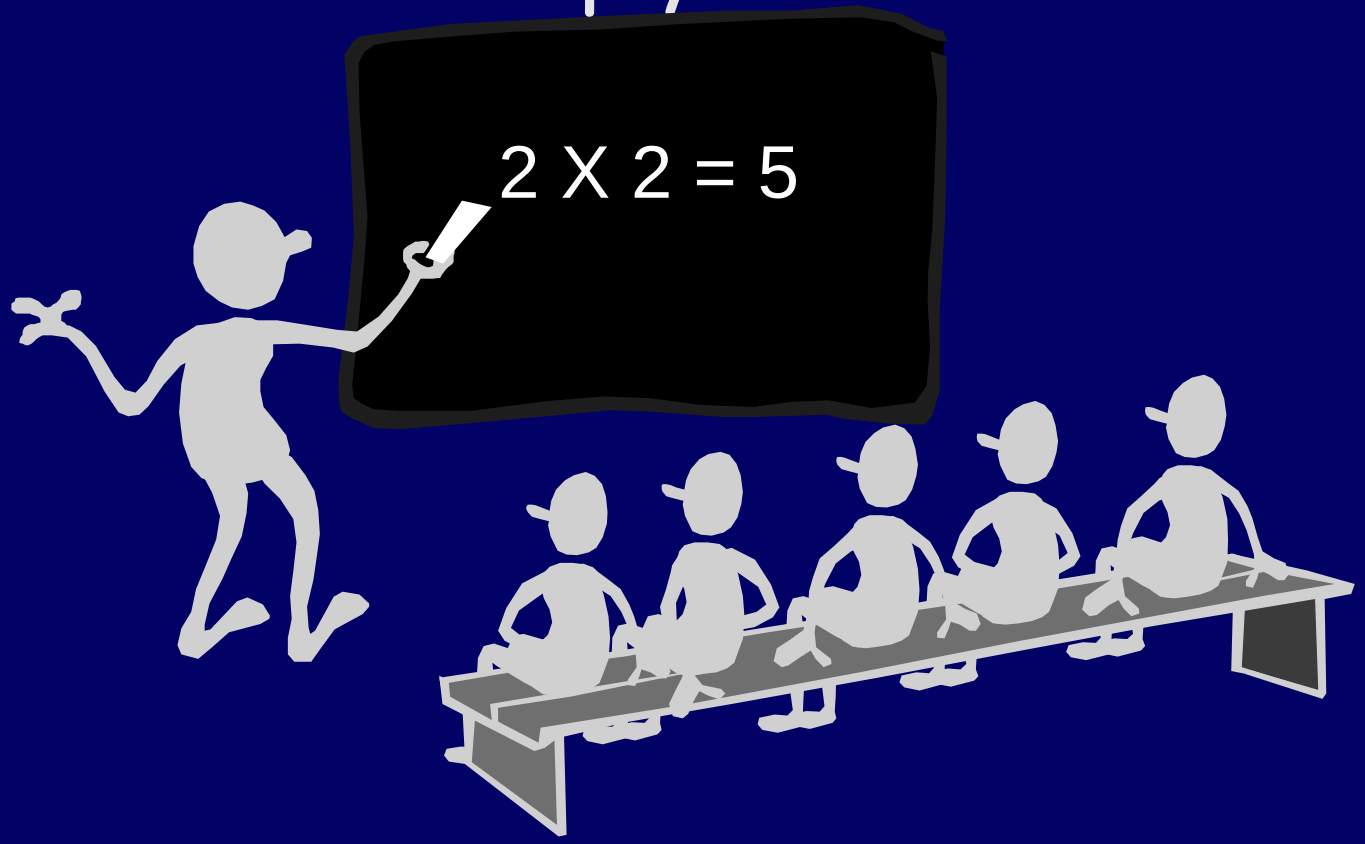
Great Theoretical Ideas In Computer Science

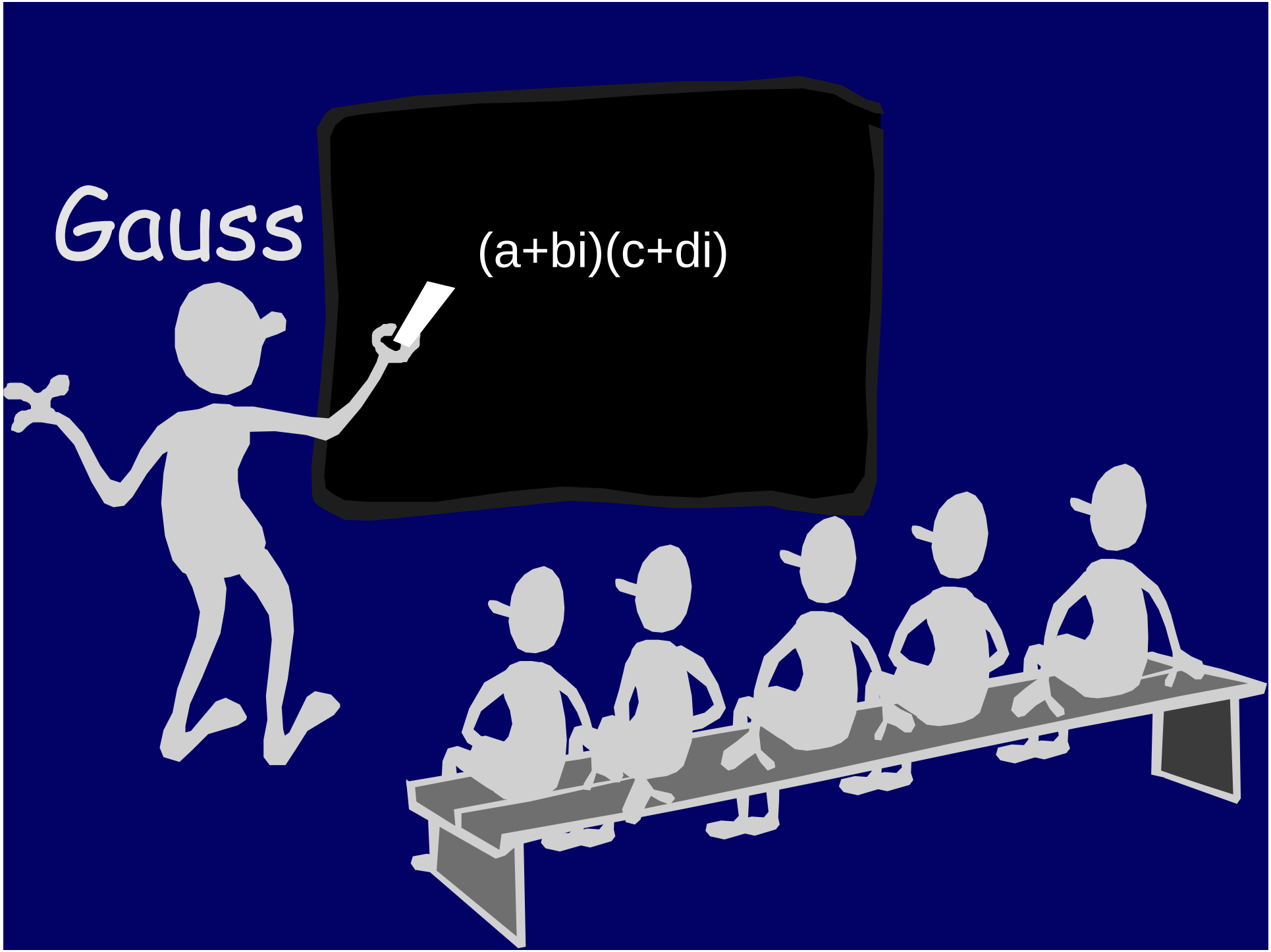
Steven Rudich CS 15-251 Spring 2004

Lecture 16 March 4, 2004 Carnegie Mellon University

Grade School Revisited: How To Multiply Two Numbers




The best way is often far from obvious!


Gauss
(a+bi)(c+di)

Gauss' Complex Puzzle

Remember how to multiply 2 complex numbers a + bi and c + di?

(a+bi)(c+di) = [ac –bd] + [ad + bc] i

Input: a,b,c,d Output: ac-bd, ad+bc

If a real multiplication costs $1 and an addition cost a penny. What is the cheapest way to obtain the output from the input?

Can you do better than $4.02?

Gauss' $3.05 Method:

Input: a,b,c,d Output: ac-bd, bc+ad

$X_1 = a + b$

$X_2 = c + d$

$X_3 = X_1 X_2 = ac + ad + bc + bd$

$X_4 = ac$

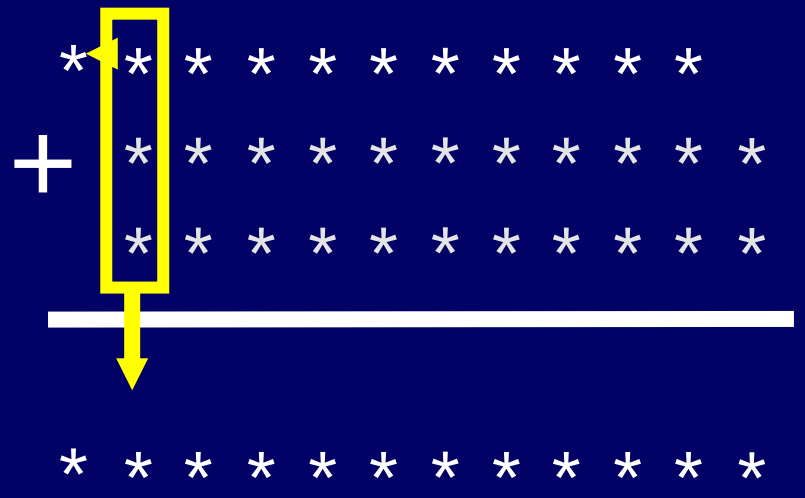
$X_5 = bd$

$X_6 = X_4 - X_5 =$ ac-bd

$X_7 = X_3 - X_4 - X_5 = bc + ad$

The Gauss optimization saves one multiplication out of four. It requires 25% less work.

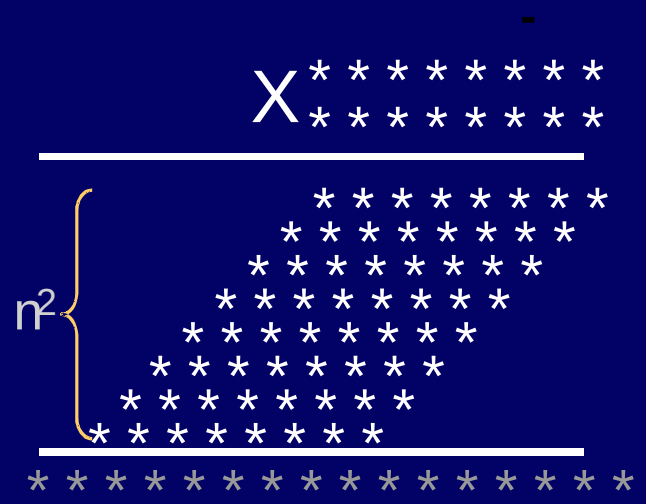
Time complexity of grade school addition



T(n) = The amount of time grade school addition uses to add two n-bit numbers

We saw that T(n) was linear.

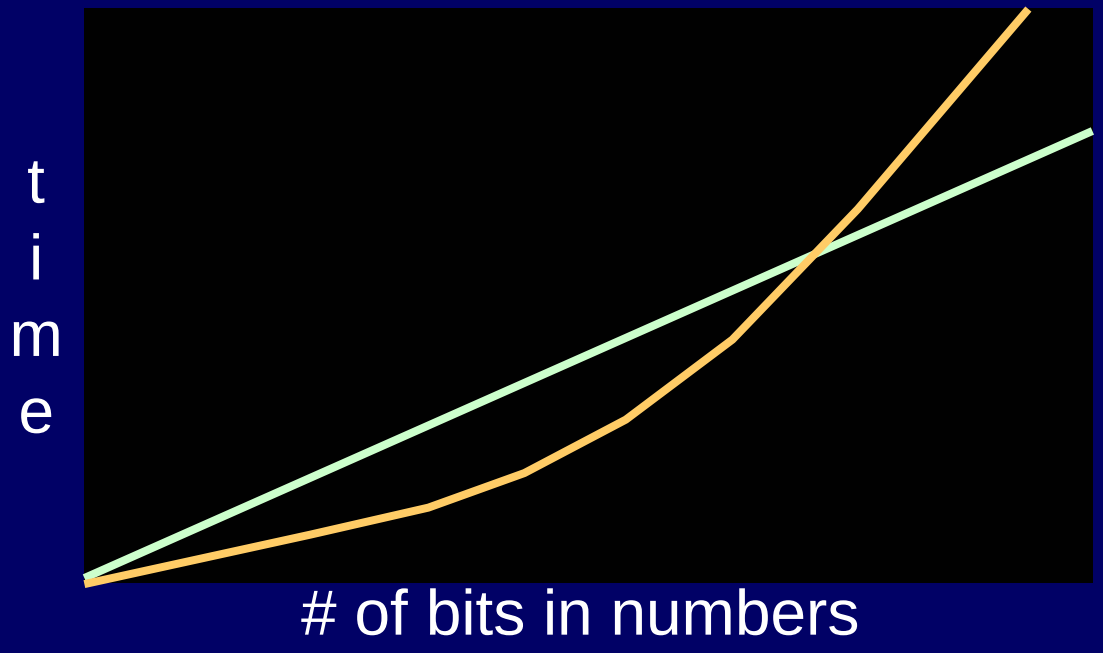
Time complexity of grade school multiplication



T(n) = The amount of time grade school multiplication uses to add two n-bit numbers

We saw that T(n) was quadratic.

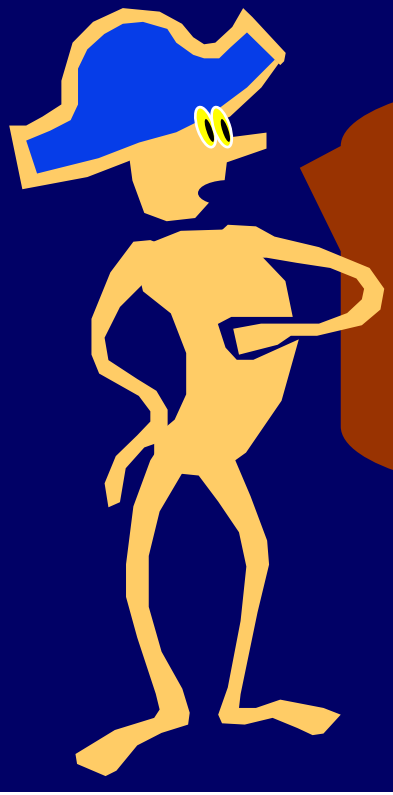
Grade School Addition: Linear time
Grade School Multiplication: Quadratic time



No matter how dramatic the difference in the constants the quadratic curve will eventually dominate the linear curve

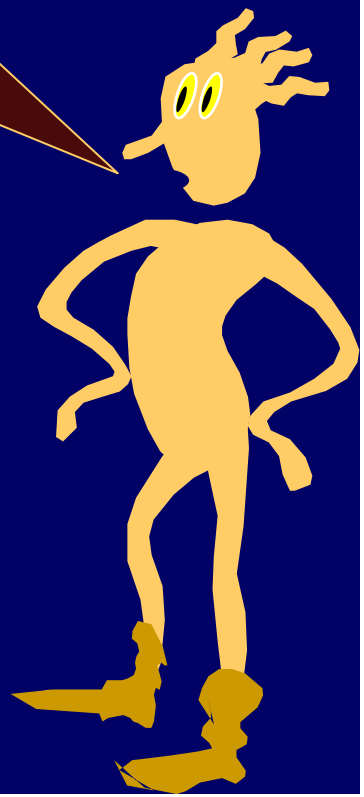
Grade School Addition: $\Theta(n)$ time
Furthermore, it is optimal

Grade School Multiplication: $\Theta(n^2)$ time



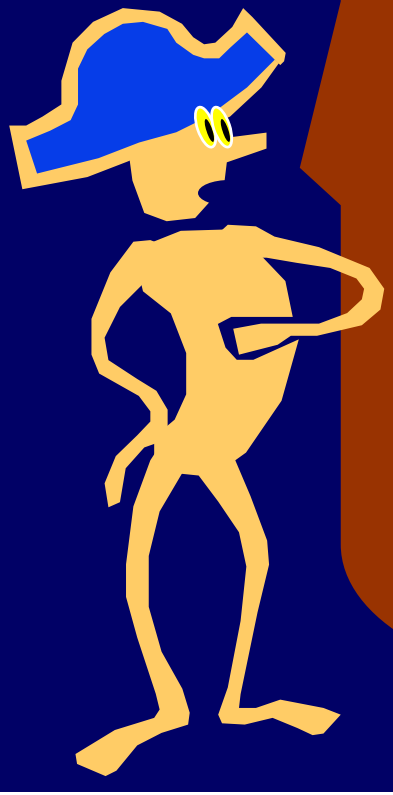
Is there a clever algorithm to multiply two numbers in linear time?

Open Problem!



Is there a faster way to multiply two numbers than the way you learned in grade school?

Grade School Multiplication: $\Theta(n^2)$ time Kissing Intuition



Intuition: Let's say that each time an algorithm has to multiply a digit from one number with a digit from the other number, we call that a "kiss". It seems as if any correct algorithm must kiss at least n^2 times.

Divide And Conquer
(an approach to faster algorithms)

DIVIDE a problem into smaller subproblems

CONQUER them recursively

GLUE the answers together so as to obtain the answer to the larger problem

Multiplication of 2 n-bit numbers

$$X = a\,2^{n/2} + b \qquad Y = c\,2^{n/2} + d$$

$$XY = ac\,2^{n} + (ad+bc)\,2^{n/2} + bd$$

Multiplication of 2 n-bit numbers

X = | a | b |

Y = | c | d |

$XY = ac\ 2^n + (ad+bc)\ 2^{n/2} + bd$

MULT(X,Y):

If |X| = |Y| = 1 then RETURN XY

Break X into a;b and Y into c;d

RETURN

$MULT(a,c)\ 2^n + (MULT(a,d) + MULT(b,c))\ 2^{n/2} + MULT(b,d)$

Multiplying (Divide & Conquer style)

12345678 * 21394276

*4276 5678*2139 5678*4276

Multiplying (Divide & Conquer style)

12345678 * 21394276

1234*2139 1234*4276 5678*2139 5678*4276

Multiplying (Divide & Conquer style)

12345678 * 21394276

1234*2139 1234*4276 5678*2139 5678*4276

Multiplying (Divide & Conquer style)

12345678 * 21394276

1234*2139 1234*4276 5678*2139 5678*4276

Multiplying (Divide & Conquer style)

12345678 * 21394276

1234*2139 1234*4276 5678*2139 5678*4276

Multiplying (Divide & Conquer style)

12345678 * 21394276

1234*2139 1234*4276 5678*2139 5678*4276

12*21 12*39 34*21 34*39 xxxxxxxxxxxxxxxxxxxxxxxx

Multiplying (Divide & Conquer style)

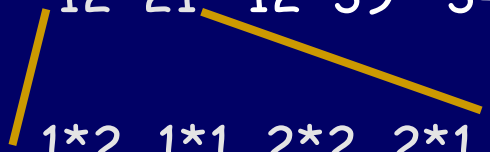
Multiplying (Divide & Conquer style)

12345678 * 21394276

1234*2139 1234*4276 5678*2139 5678*4276

12*21 12*39 34*21 34*39 xxxxxxxxxxxxxxxxxxxxxxxx

1*2 1*1 2*2 2*1 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx



Multiplying (Divide & Conquer style)

12345678 * 21394276

1234*2139 1234*4276 5678*2139 5678*4276

12*21 12*39 34*21 34*39 xxxxxxxxxxxxxxxxxxxxxxxx

2 1 4 2 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Multiplying (Divide & Conquer style)

12345678 * 21394276

1234*2139 1234*4276 5678*2139 5678*4276

12*21 12*39 34*21 34*39 xxxxxxxxxxxxxxxxxxxxxxxx

2 1 4 2 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Hence: $12*21 = 2*10^2 + (1 + 4)10^1 + 2 = 252$

Multiplying (Divide & Conquer style)

12345678 * 21394276

1234*2139 1234*4276 5678*2139 5678*4276

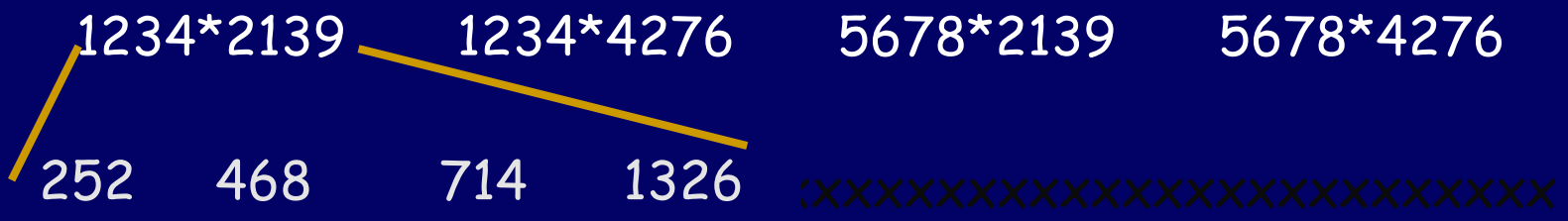
252 12*39 34*21 34*39 xxxxxxxxxxxxxxxxxxxxxxxx

2 1 4 2 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Hence: $12*21 = 2*10^2 + (1 + 4)10^1 + 2 = 252$

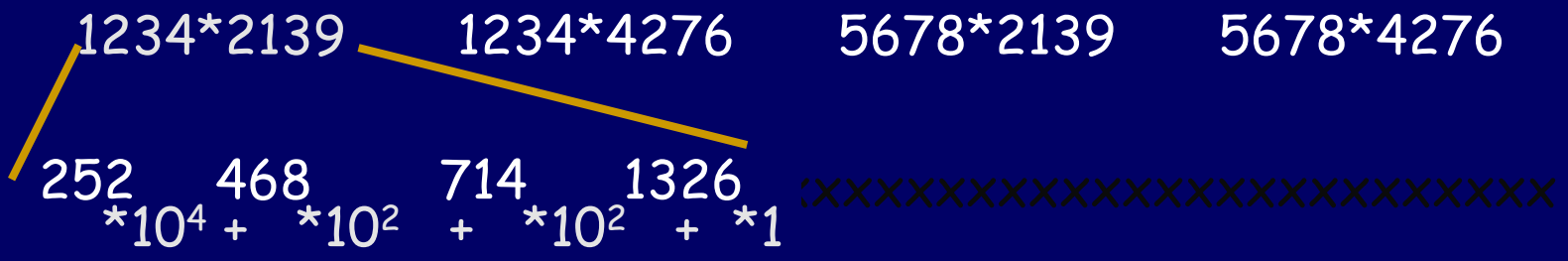
Multiplying (Divide & Conquer style)

12345678 * 21394276



Multiplying (Divide & Conquer style)

12345678 * 21394276



Multiplying (Divide & Conquer style)

12345678 * 21394276

1234*2139 1234*4276 5678*2139 5678*4276

252 468 714 1326 :xxxxxxxxxxxxxxxxxxxxxxxx

$*10^4 + *10^2 + *10^2 + *1$

= 2639526

Multiplying (Divide & Conquer style)

12345678 * 21394276

2639526 1234*4276 5678*2139 5678*4276

Multiplying (Divide & Conquer style)

12345678 * 21394276

2639526 5276584 12145242 24279128

Multiplying (Divide & Conquer style)

$$12345678 * 21394276$$

$$2639526 * 10^8 + 5276584 * 10^4 + 12145242 * 10^4 + 24279128 * 1$$

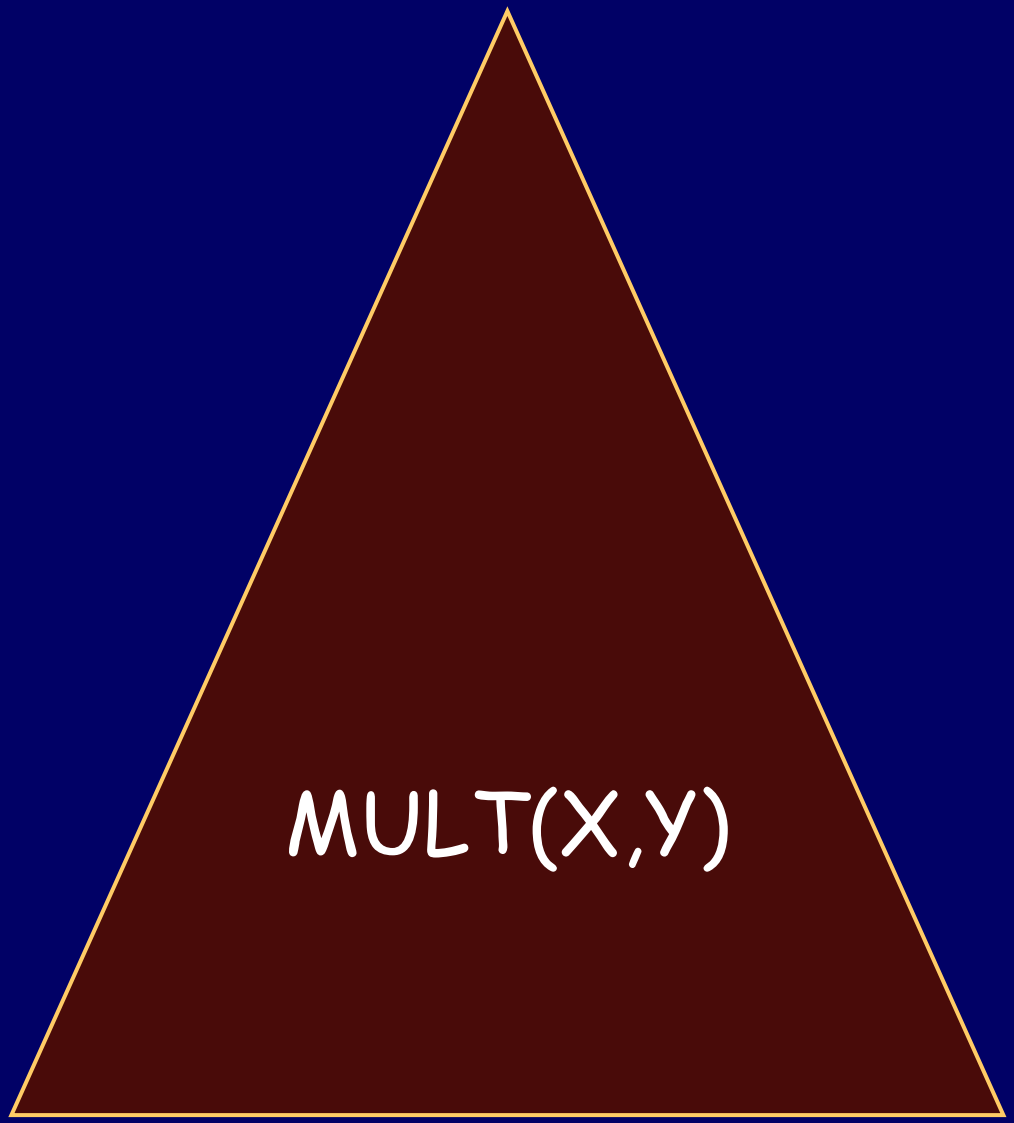
Multiplying (Divide & Conquer style)

12345678 * 21394276

$$2639526 * 10^8 + 5276584 * 10^4 + 12145242 * 10^4 + 24279128 * 1$$

= 264126842539128

Divide, Conquer, and Glue



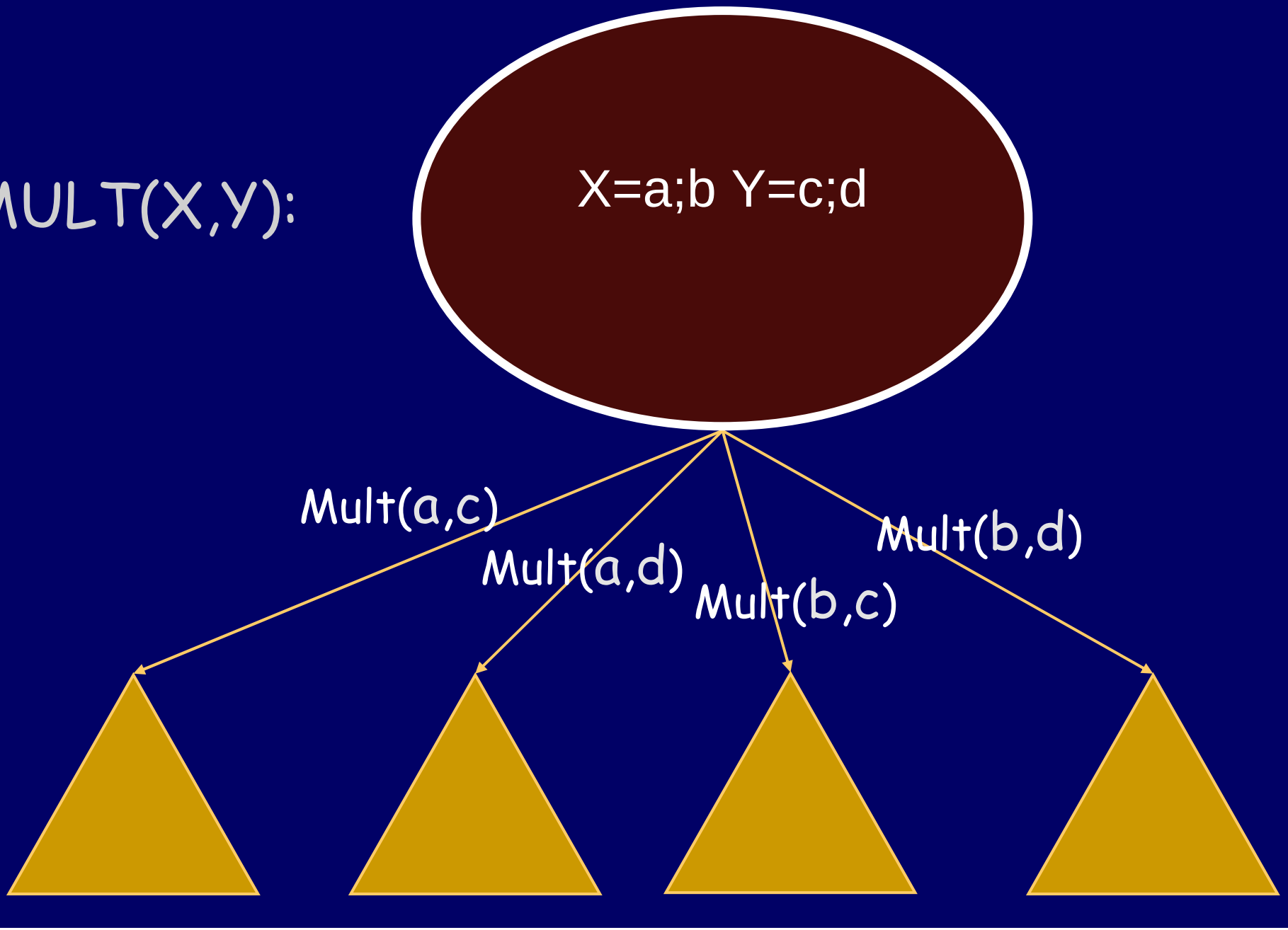
Divide, Conquer, and Glue

MULT(X,Y):

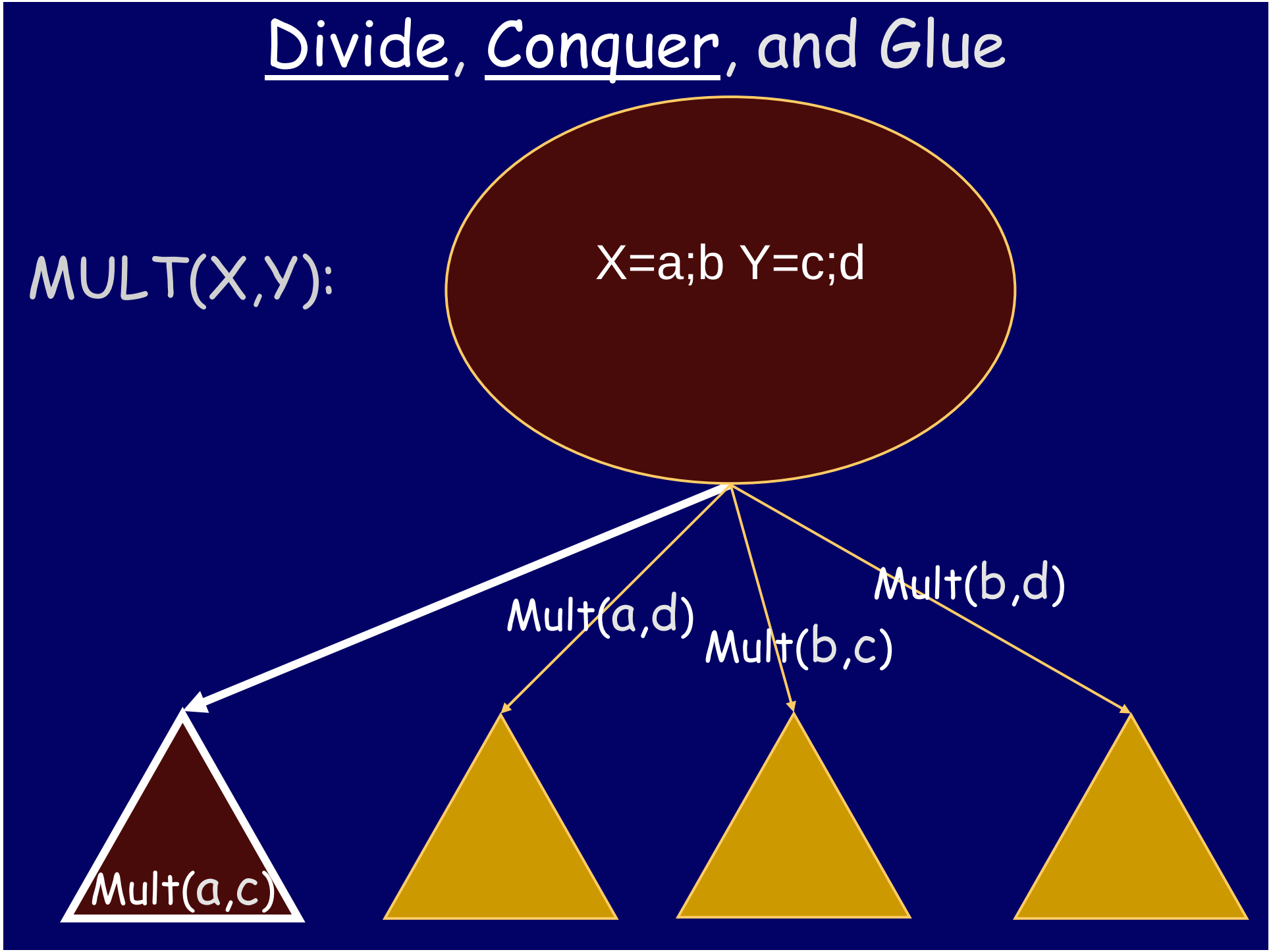
If |X|=|Y|=1,
Then Return XY
Else ….

Divide, Conquer, and Glue

MULT(X,Y):

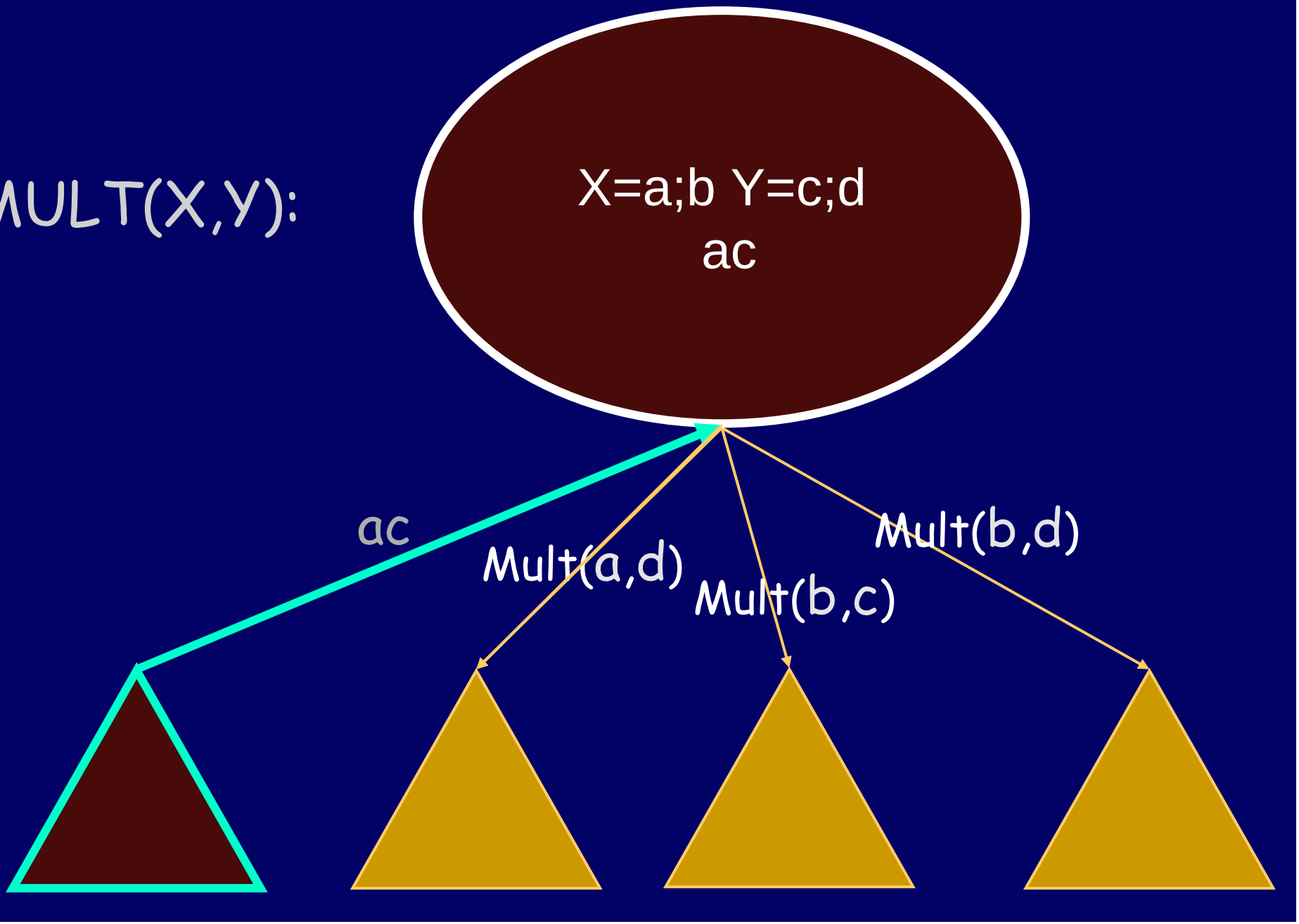


Divide, Conquer, and Glue

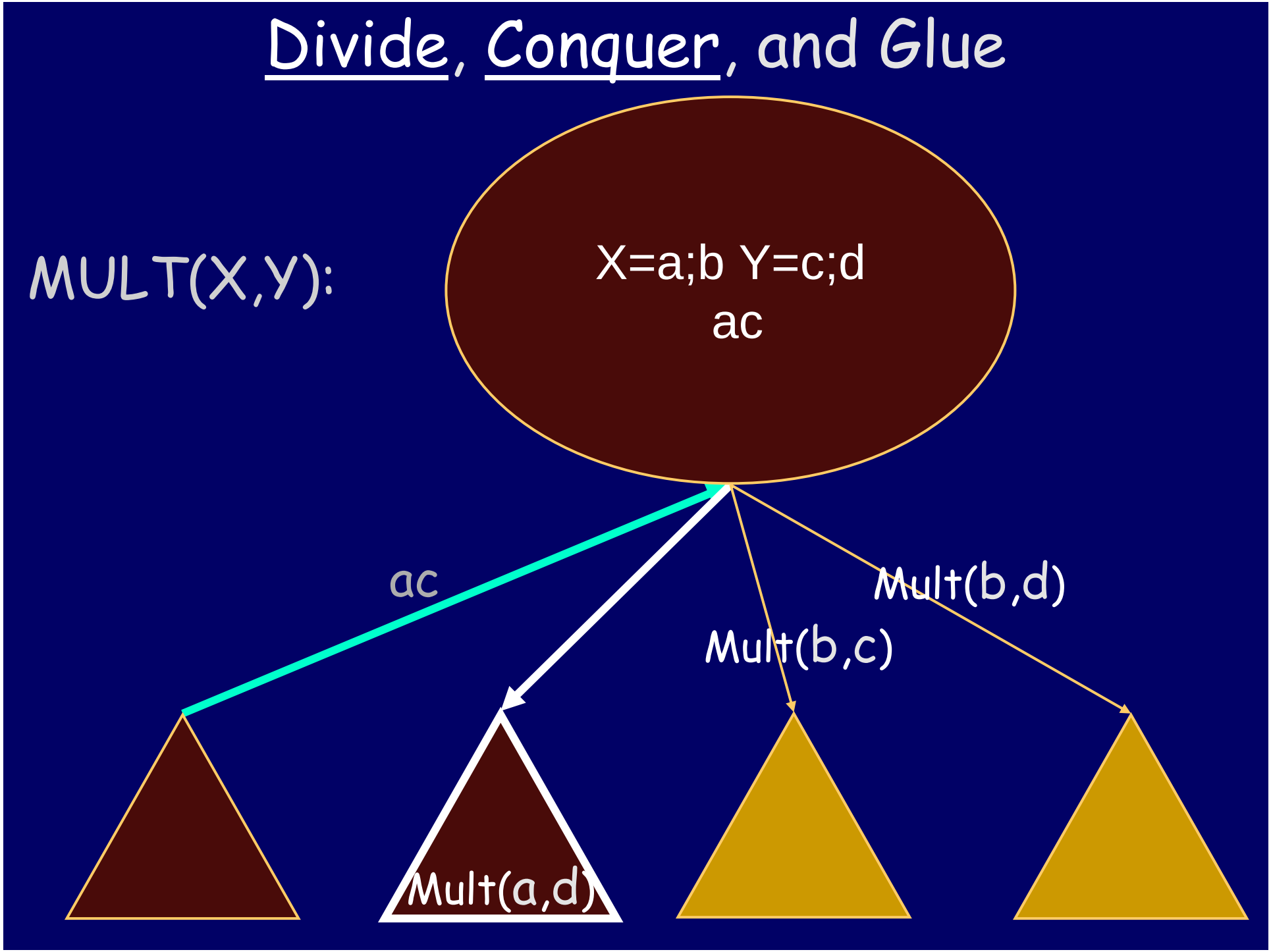


Divide, Conquer, and Glue

MULT(X,Y):

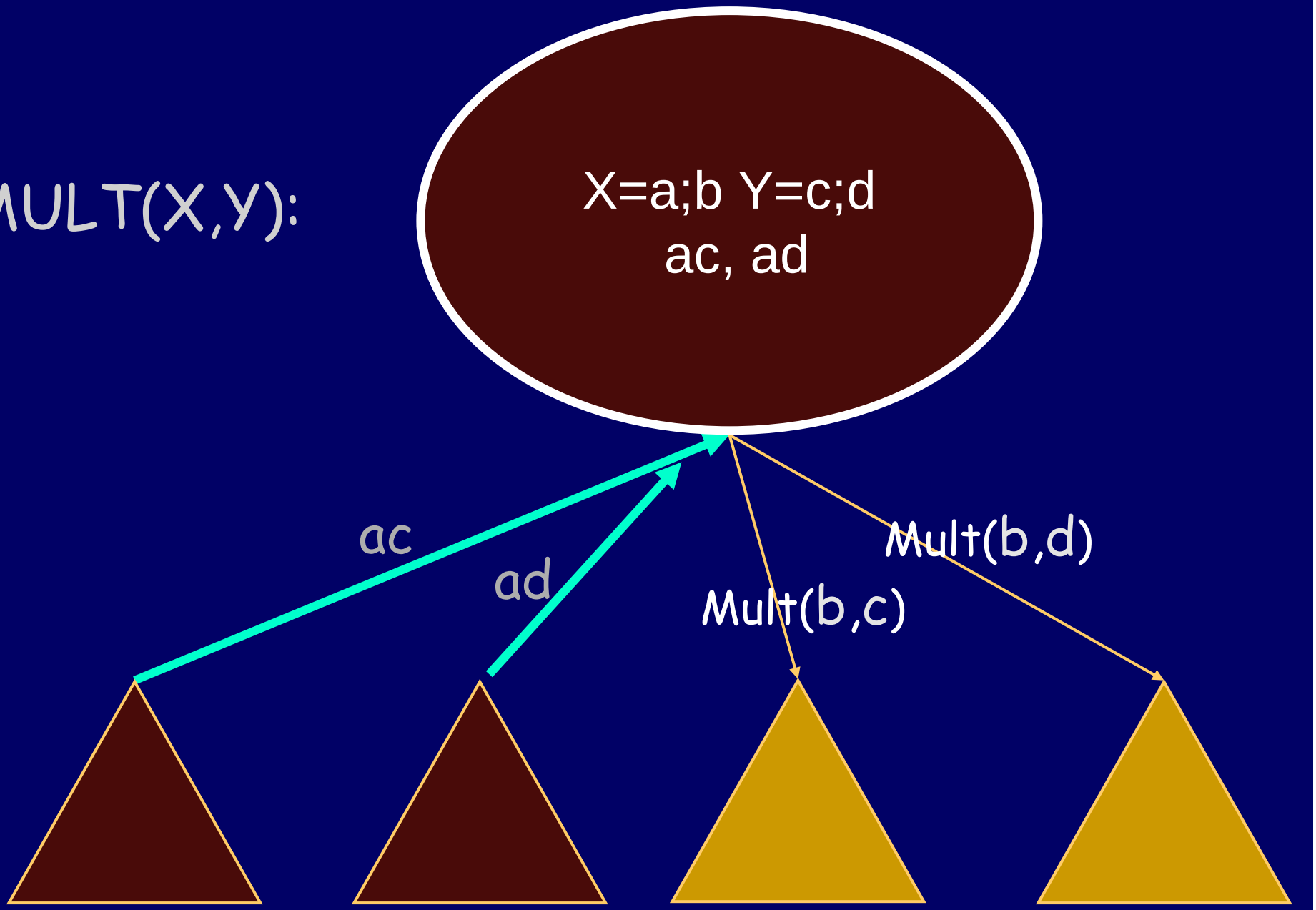


Divide, Conquer, and Glue



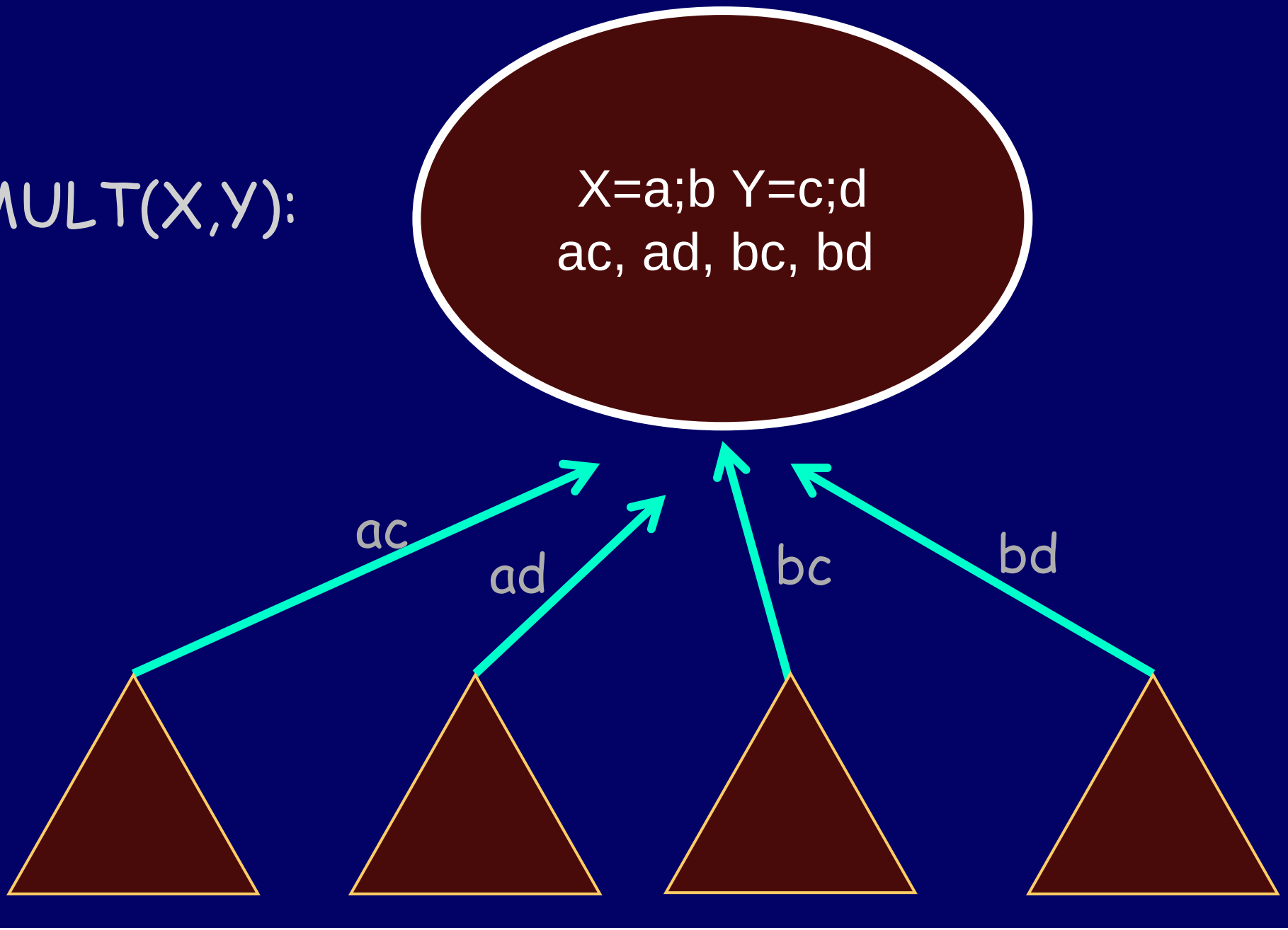
Divide, Conquer, and Glue

MULT(X,Y):



Divide, Conquer, and Glue

MULT(X,Y):

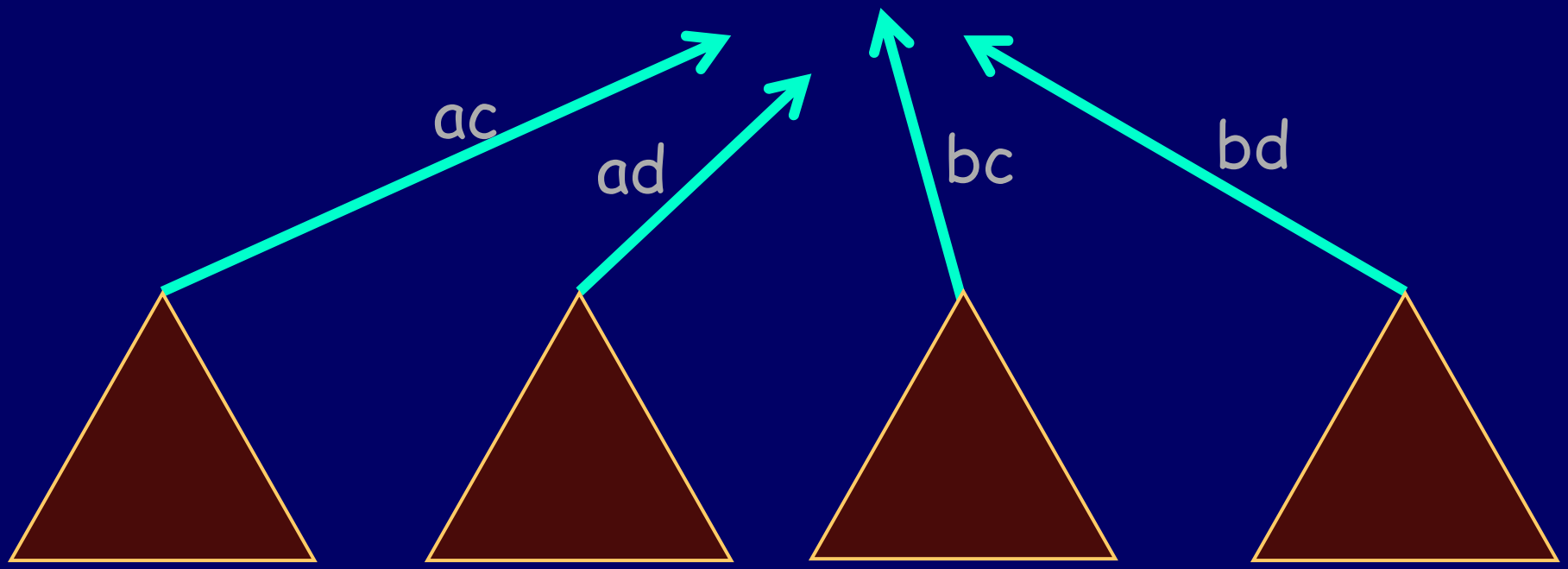


Divide, Conquer, and Glue

MULT(X,Y):

X=a;b Y=c;d

$ac2^n + (ad+bc)2^{n/2} + bd$



Time required by MULT

T(n) = time taken by MULT on two n-bit numbers

What is T(n)? What is its growth rate? Is it $\theta(n^2)$?

$T(n) = 4\,T(n/2) + k'\,n + k''\,, \quad n=|X|$

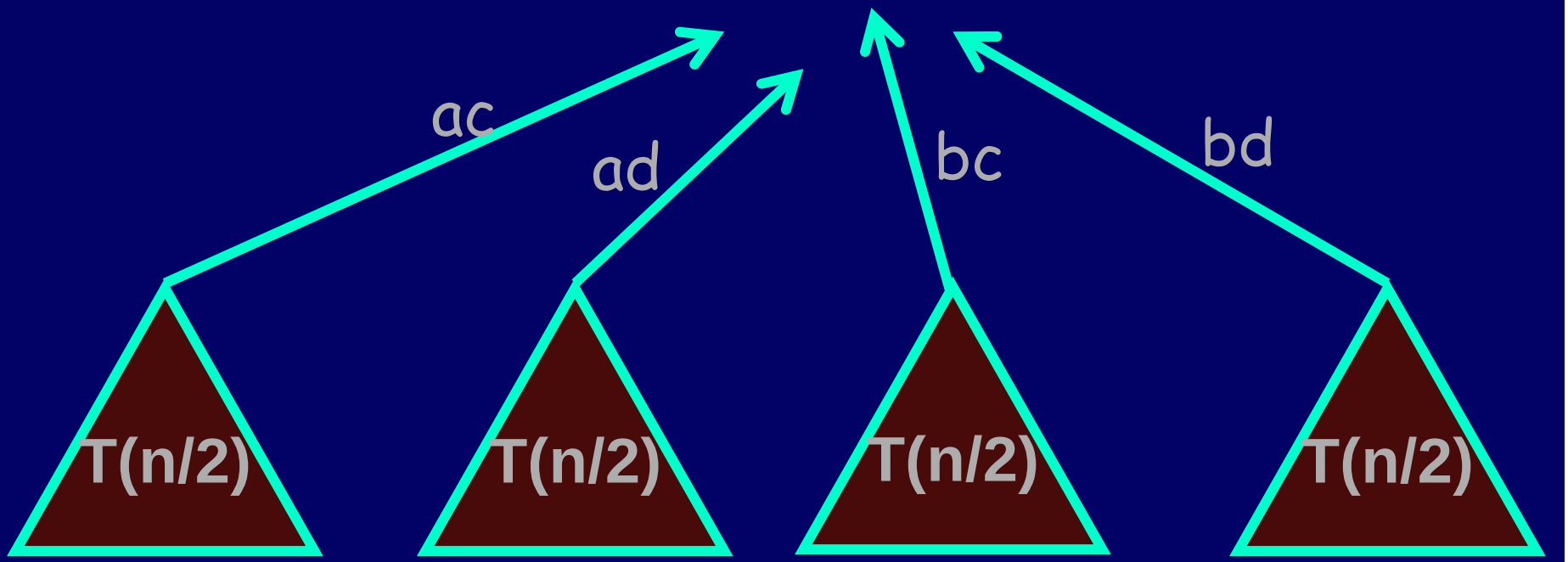
Conquer

Divide and glue

X=a;b Y=c;d

$ac2^n + (ad+bc)2^{n/2} + bd$

Time: k'n + k"



Recurrence Relation

$T(1) = k$ for some constant k

$T(n) = 4\ T(n/2) + k' n + k''$ for some constants k' and k''

MULT(X,Y):

If |X| = |Y| = 1 then RETURN XY

Break X into a;b and Y into c;d

RETURN

$MULT(a,c)\ 2^n + (MULT(a,d) + MULT(b,c))\ 2^{n/2} + MULT(b,d)$

Let's be concrete to keep it simple

$T(1) = 1$

$T(n) = 4\ T(n/2) + n$

Notice that T(n) is inductively defined only for positive powers of 2

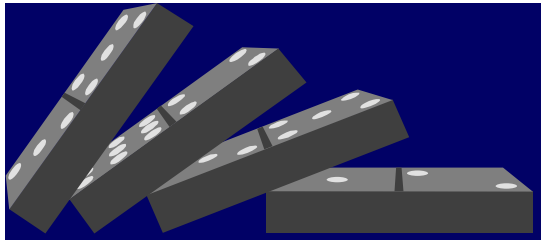
How do we unravel T(n) so that we can determine its growth rate?

Technique 1
Guess and Verify

Guess: $G(n) = 2n^2 - n$

Verify: $G(1) = 1$ and $G(n) = 4\ G(n/2) + n$

$$
\begin{aligned}
& 4\,[2(n/2)^2 - n/2] + n \\
= \quad & 2n^2 - 2n + n \\
= \quad & 2n^2 - n \\
= \quad & G(n)
\end{aligned}
$$



Technique 1
Guess and Verify

Guess: $G(n) = 2n^2 - n$

Verify: $G(1) = 1$ and $G(n) = 4\,G(n/2) + n$

Similarly: $T(1) = 1$ and $T(n) = 4\,T(n/2) + n$

So $T(n) = G(n) = \theta(n^2)$

Technique 2
Guess Form and Calculate Coefficients

Guess: $T(n) = an^2 + bn + c$ for some a,b,c

Calculate: $T(1) = 1 \Rightarrow a + b + c = 1$

$$T(n) = 4\,T(n/2) + n$$

$$\Rightarrow\ an^2 + bn + c = 4\,[a(n/2)^2 + b(n/2) + c] + n$$

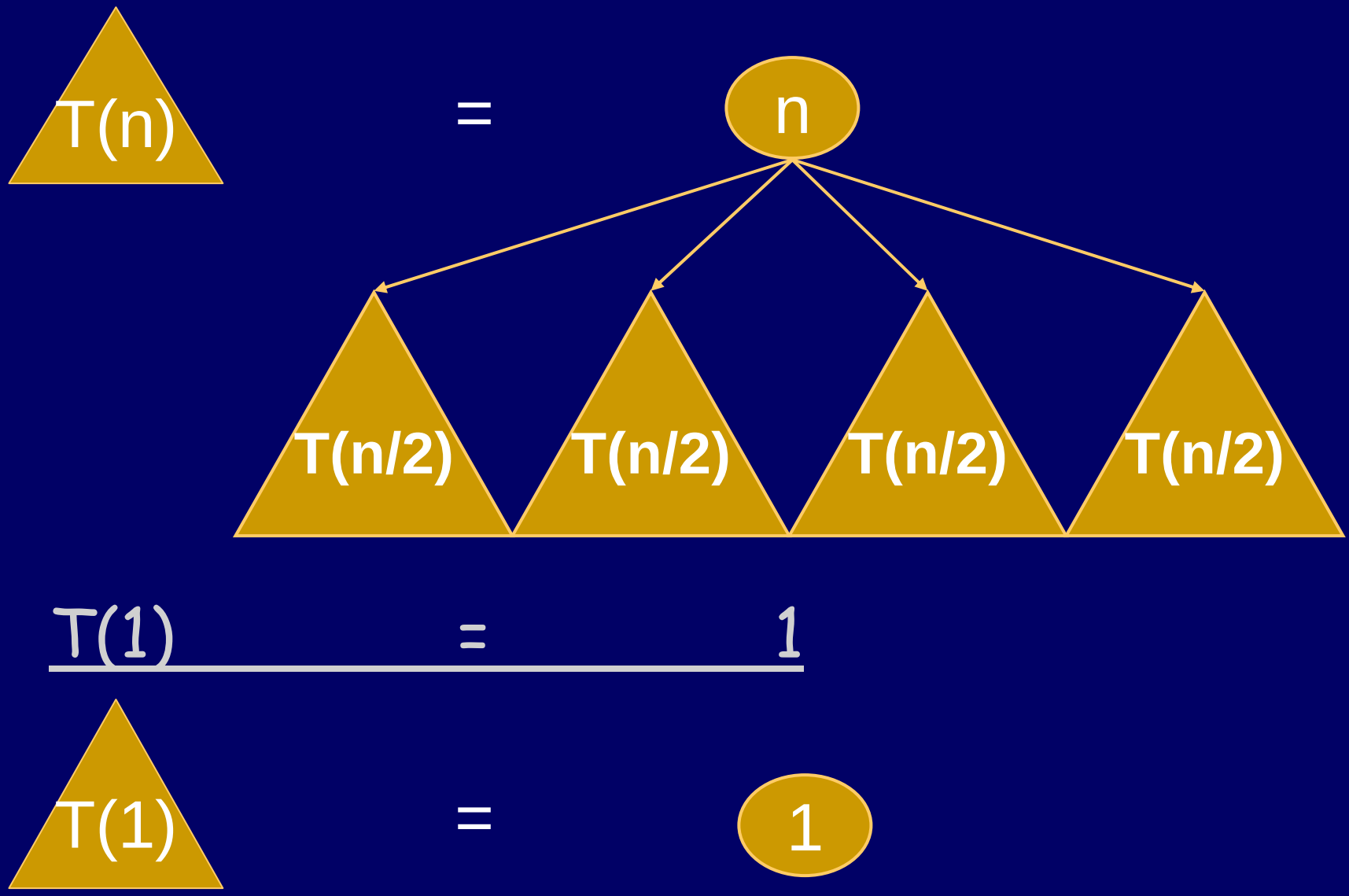
$$= an^2 + 2bn + 4c + n$$

$$\Rightarrow\ (b+1)n + 3c = 0$$

Therefore: b=-1 c=0 a=2

Technique 3: Labeled Tree Representation

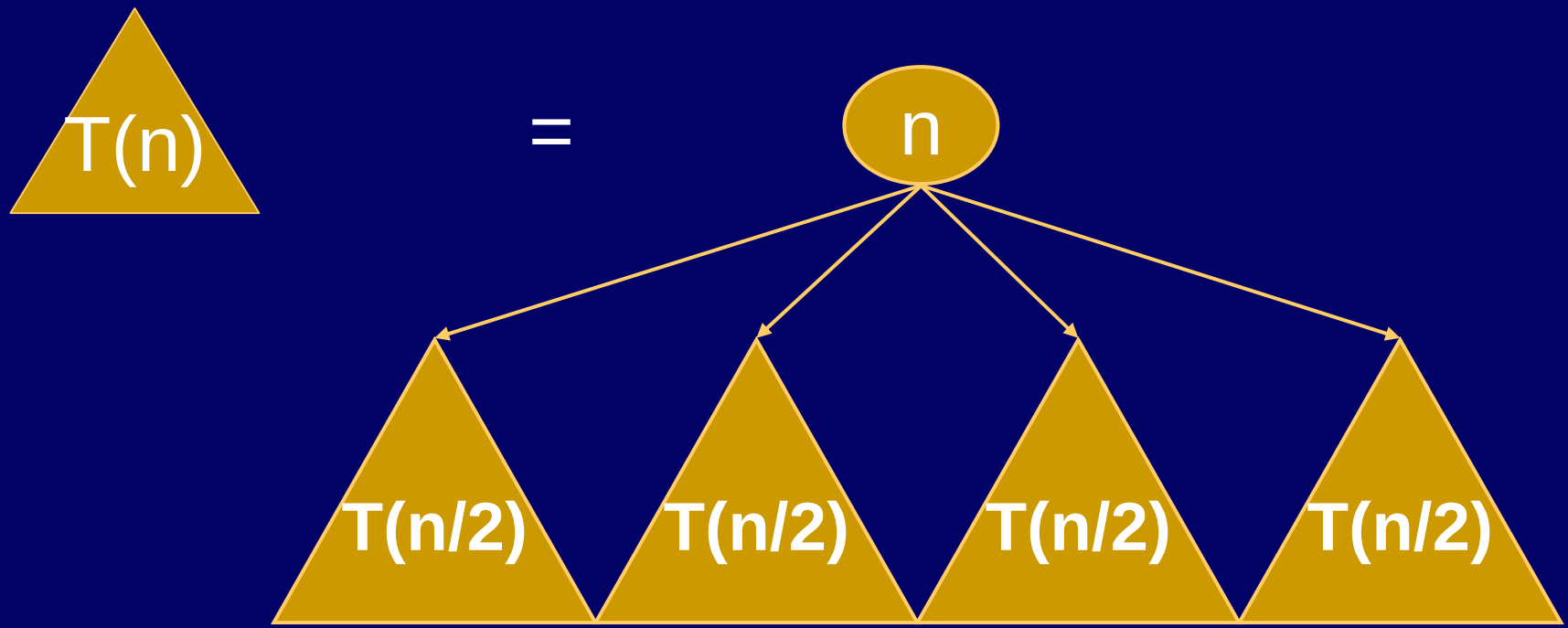
T(n) = n + 4 T(n/2)



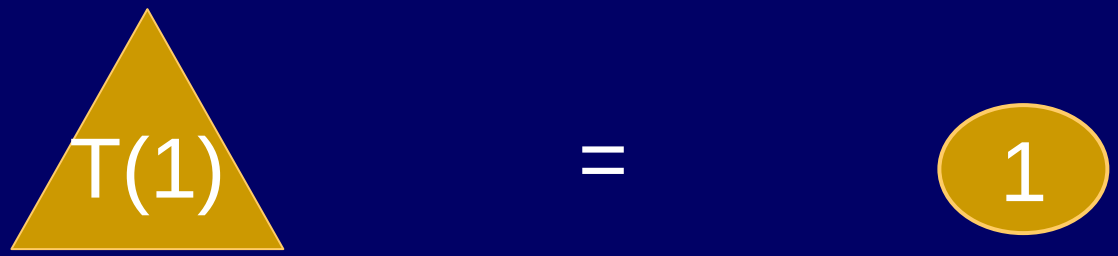
T(1) = 1

Node labels: time spent NOT conquering.

T(n) = n + 4 T(n/2)



T(1) = 1



$T(n) = 4\,T(n/2) + k'\,n + k''\,, \quad n=|X|$

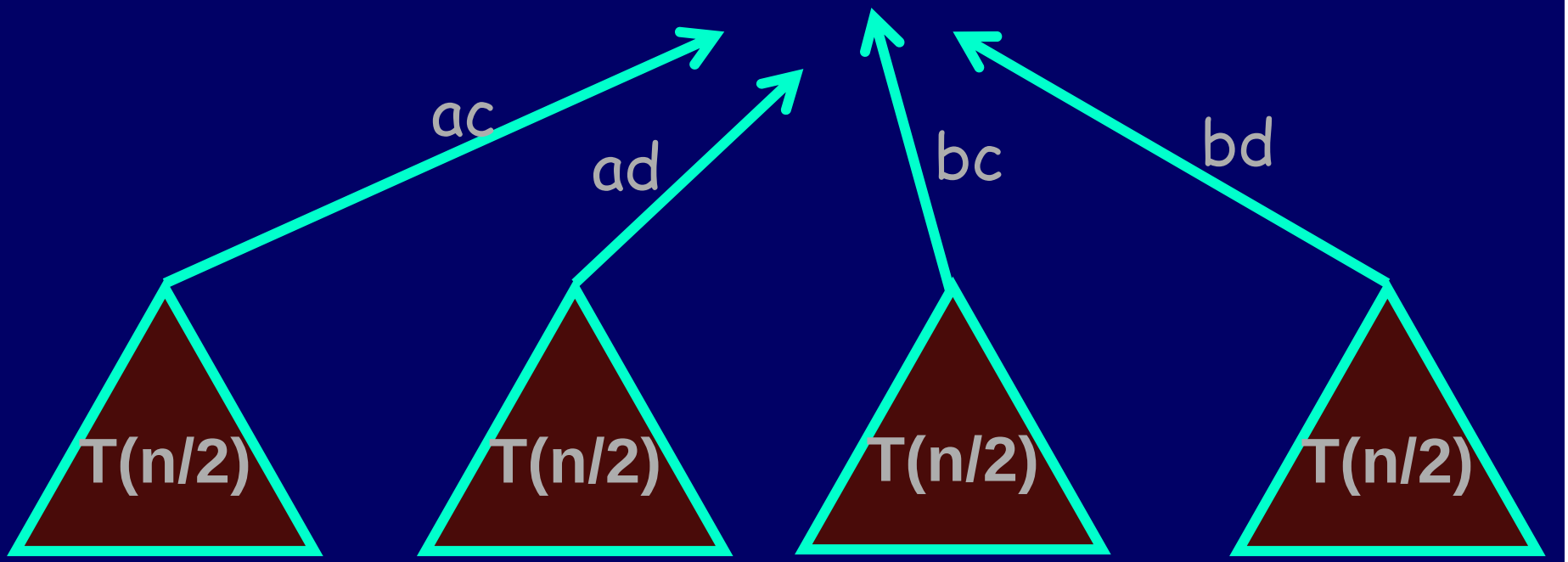
Conquer

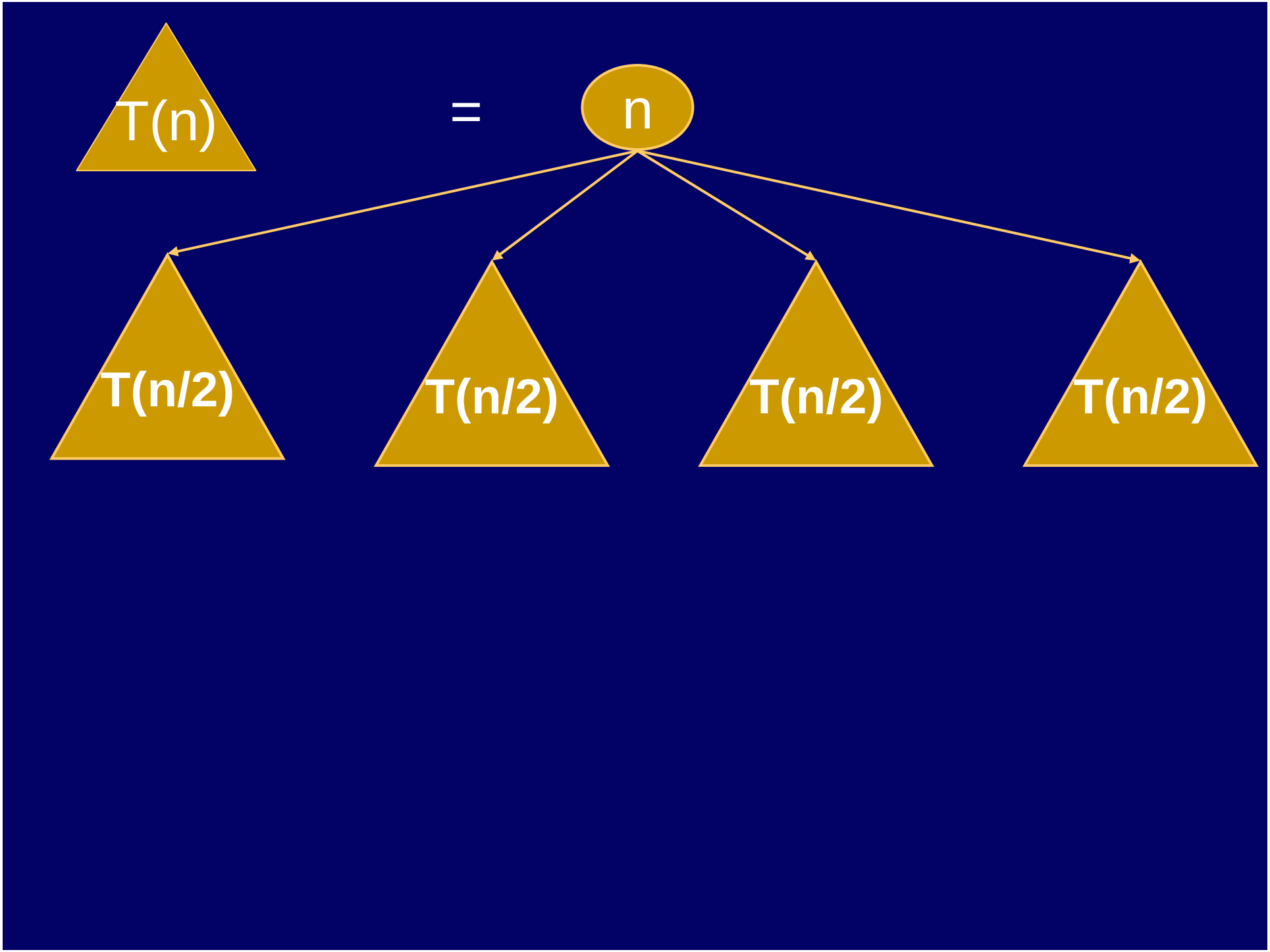
Divide and glue

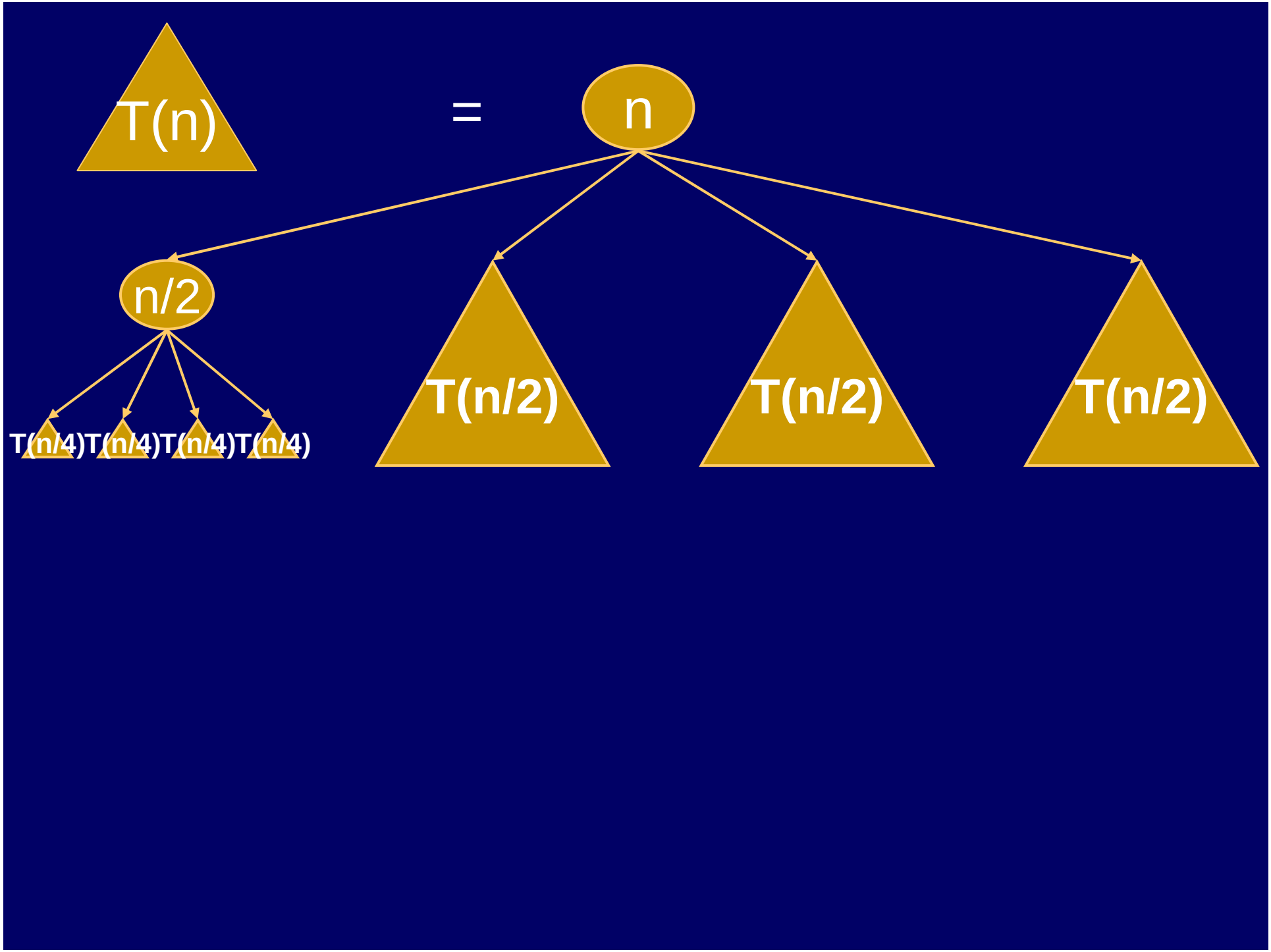
X=a;b Y=c;d

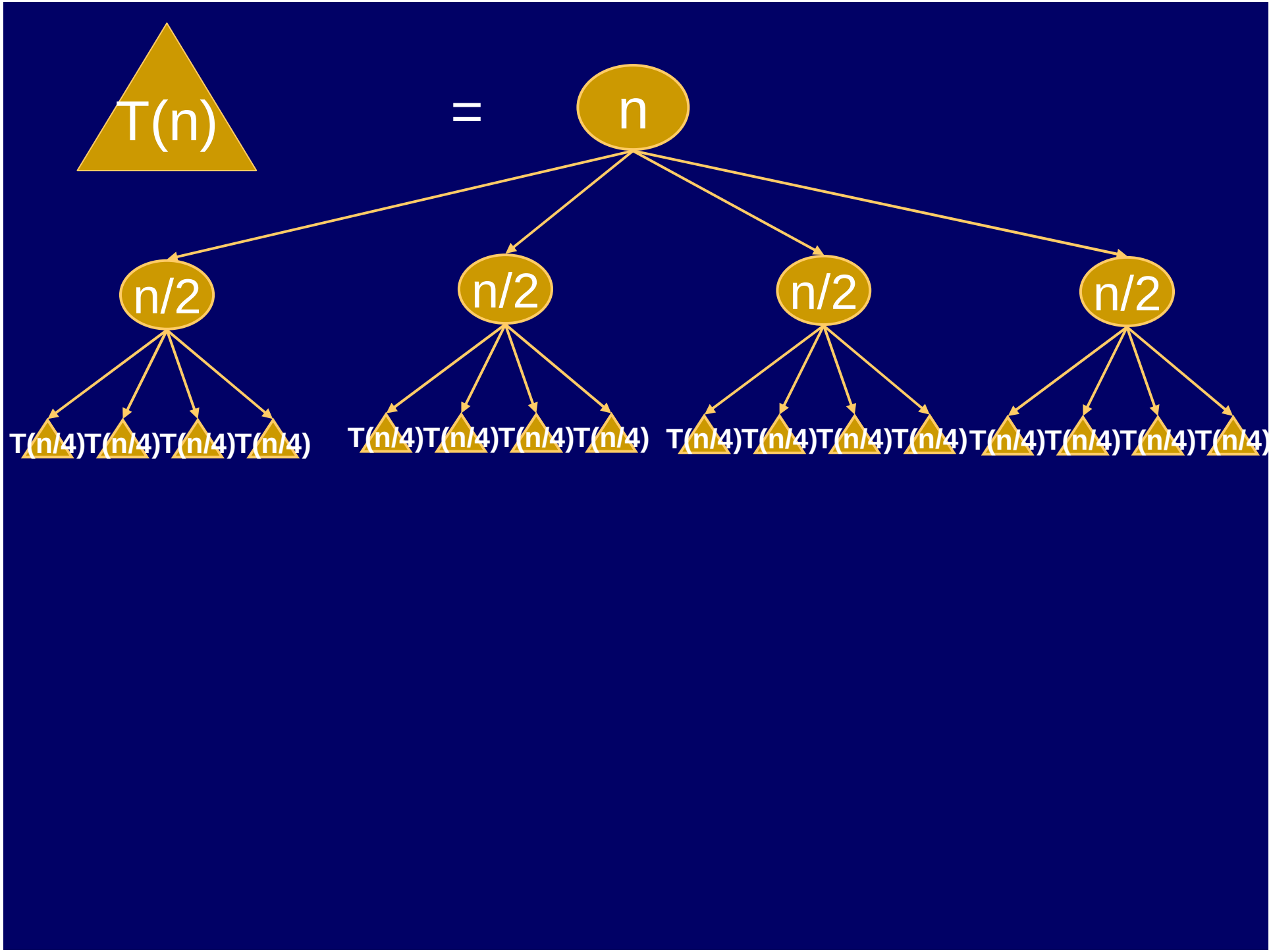
$ac2^n + (ad+bc)2^{n/2} + bd$

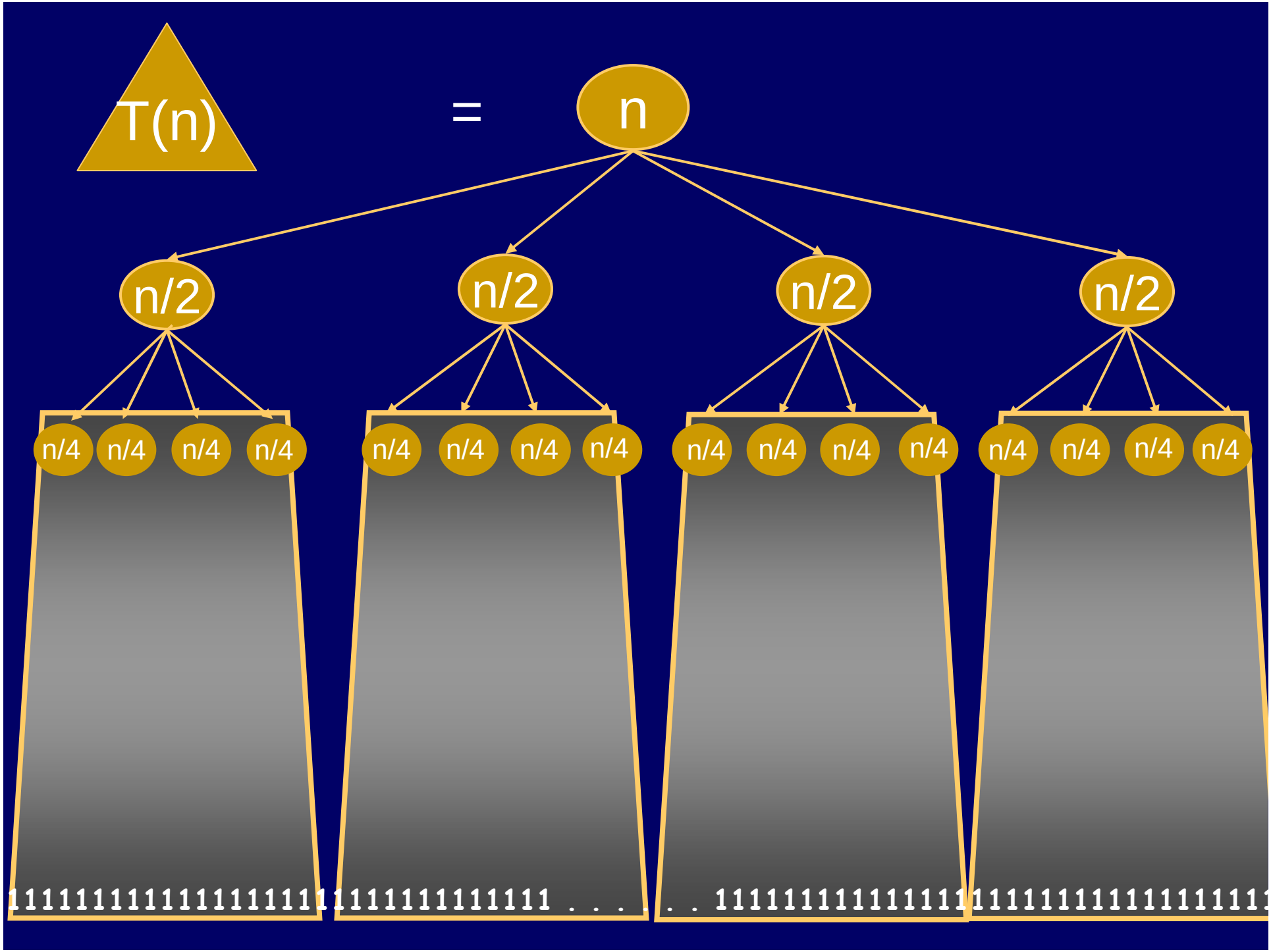
Time: k'n + k"

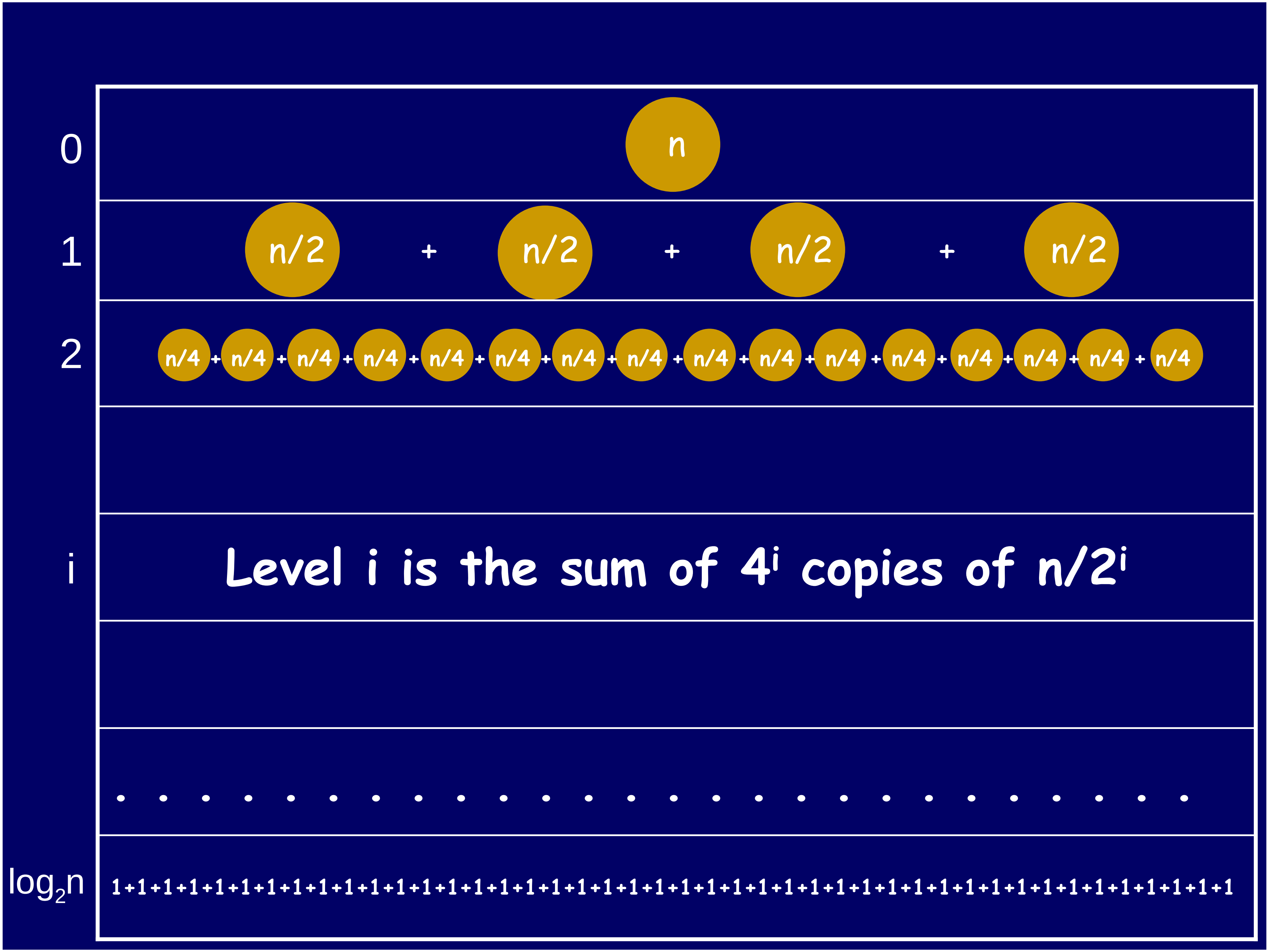


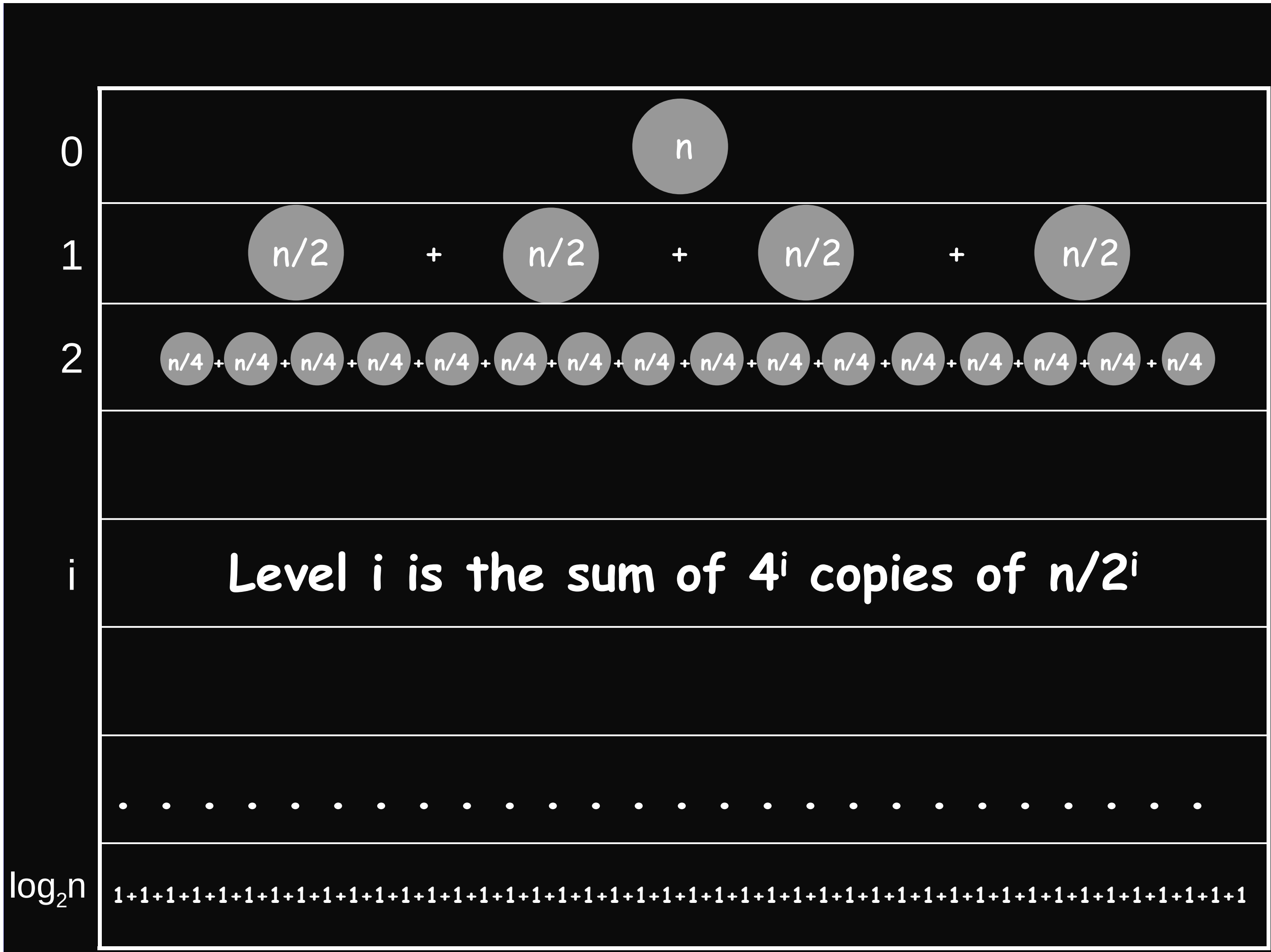

T(n)
=
n
T(n/2)
T(n/2)
T(n/2)
T(n/2)


T(n)
=
n
n/2
T(n/2)
T(n/2)
T(n/2)
T(n/4)T(n/4)T(n/4)T(n/4)


T(n)
=
n
n/2
n/2
n/2
n/2
T(n/4)T(n/4)T(n/4)T(n/4)
T(n/4)T(n/4)T(n/4)T(n/4)
T(n/4)T(n/4)T(n/4)T(n/4)
T(n/4)T(n/4)T(n/4)T(n/4)


T(n)
=
n
n/2
n/2
n/2
n/2
n/4
n/4
n/4
n/4
n/4
n/4
n/4
n/4
n/4
n/4
n/4
n/4
n/4
n/4
n/4
n/4


0
n
1
n/2 + n/2 + n/2 + n/2
2
n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4
i
Level i is the sum of 4^i copies of $n/2^i$
. . . .
$\log_2 n$
1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1



$= 1n$

$= 2n$

$= 4n$

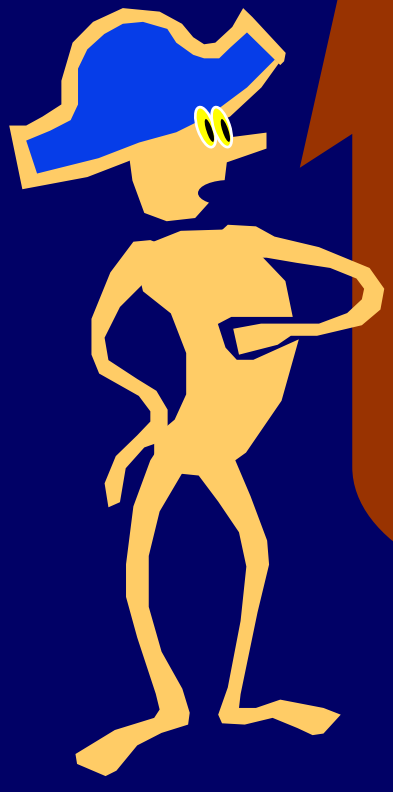
$= 2^i n$

$= (n)n$

$n(1+2+4+8+ . . . +n) =$ $n(2n-1) =$ $2n^2-n$

Divide and Conquer MULT: $\Theta(n^2)$ time
Grade School Multiplication: $\Theta(n^2)$ time

Divide and Conquer MULT: $\Theta(n^2)$ time
Grade School Multiplication: $\Theta(n^2)$ time



In retrospect, it is obvious that the kissing number for Divide and Conquer MULT is n^2, since the leaves are in correspondence with the kisses.

MULT revisited

MULT(X,Y):

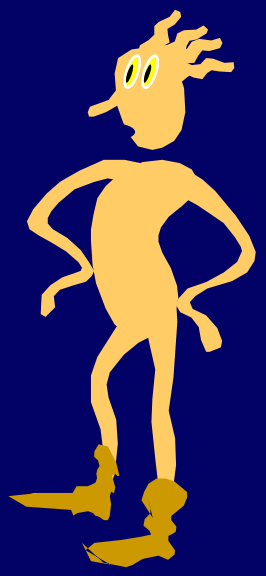
If |X| = |Y| = 1 then RETURN XY

Break X into a;b and Y into c;d

RETURN

$$MULT(a,c)\ 2^{n} + (MULT(a,d) + MULT(b,c))\ 2^{n/2} + MULT(b,d)$$

MULT calls itself 4 times.
Can you see a way to reduce the number of calls?



Gauss' Optimization:

Input: a,b,c,d Output: ac, ad+bc, bd

$X_1 = a + b$

$X_2 = c + d$

$X_3 = X_1 X_2 = ac + ad + bc + bd$

$X_4 = ac$

$X_5 = bd$

$X_6 = X_3 - X_4 - X_5 = ad + bc$

Karatsuba, Anatolii Alexeevich (1937-)

Sometime in the late 1950's Karatsuba had formulated the first algorithm to break the kissing barrier!

Gaussified MULT (Karatsuba 1962)

MULT(X,Y):

If |X| = |Y| = 1 then RETURN XY

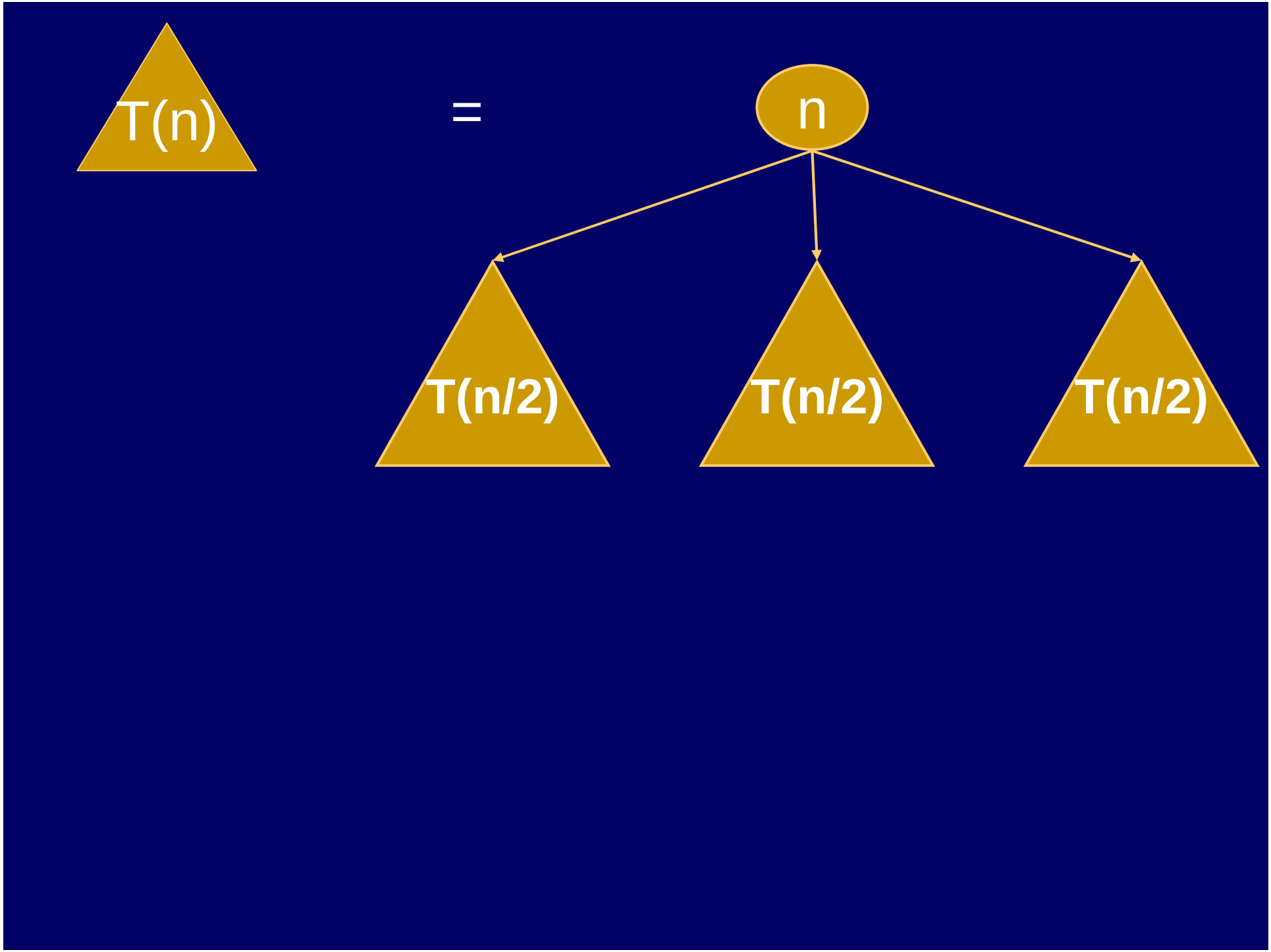
Break X into a;b and Y into c;d

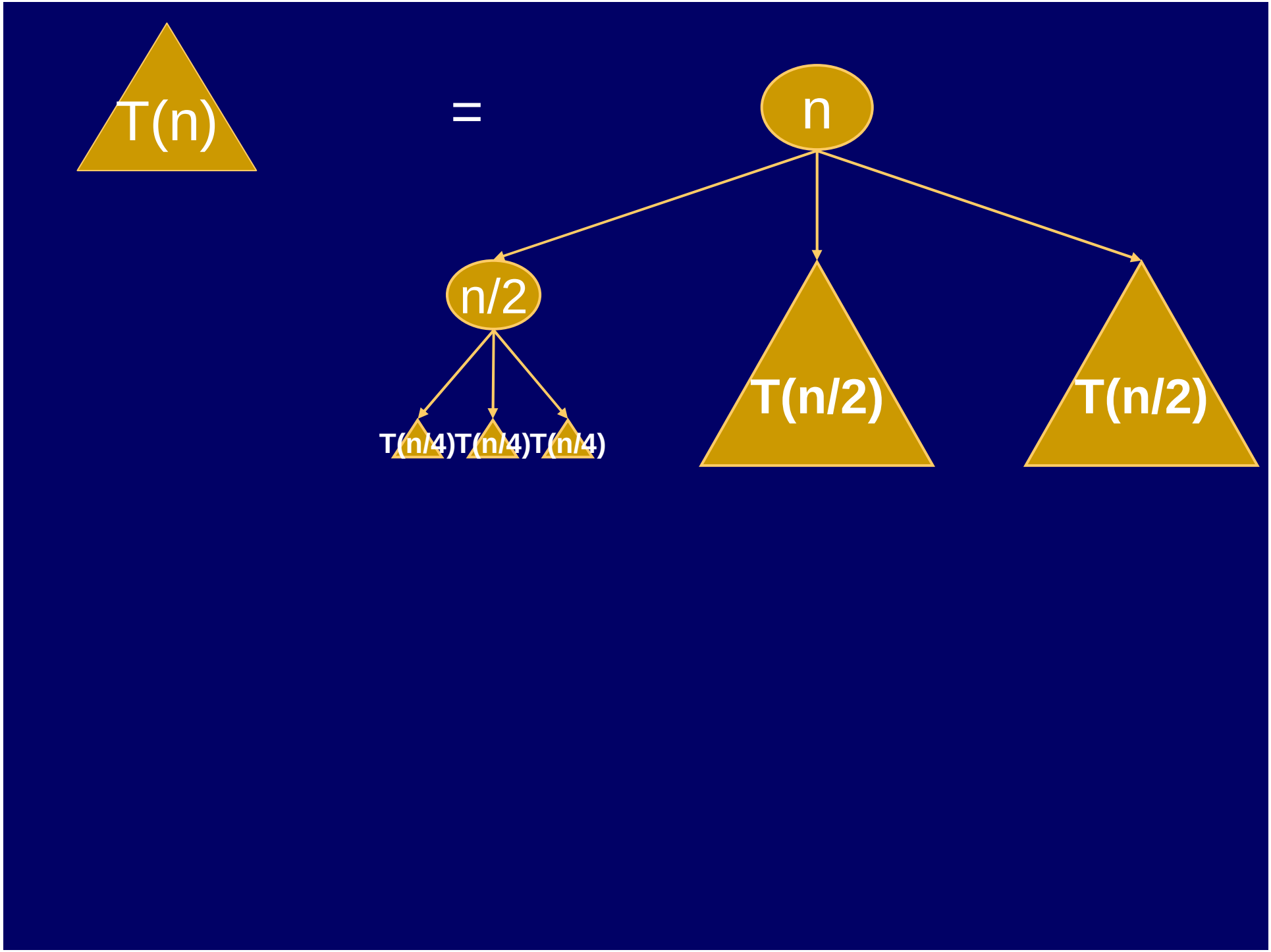
e = MULT(a,c) and f =MULT(b,d)

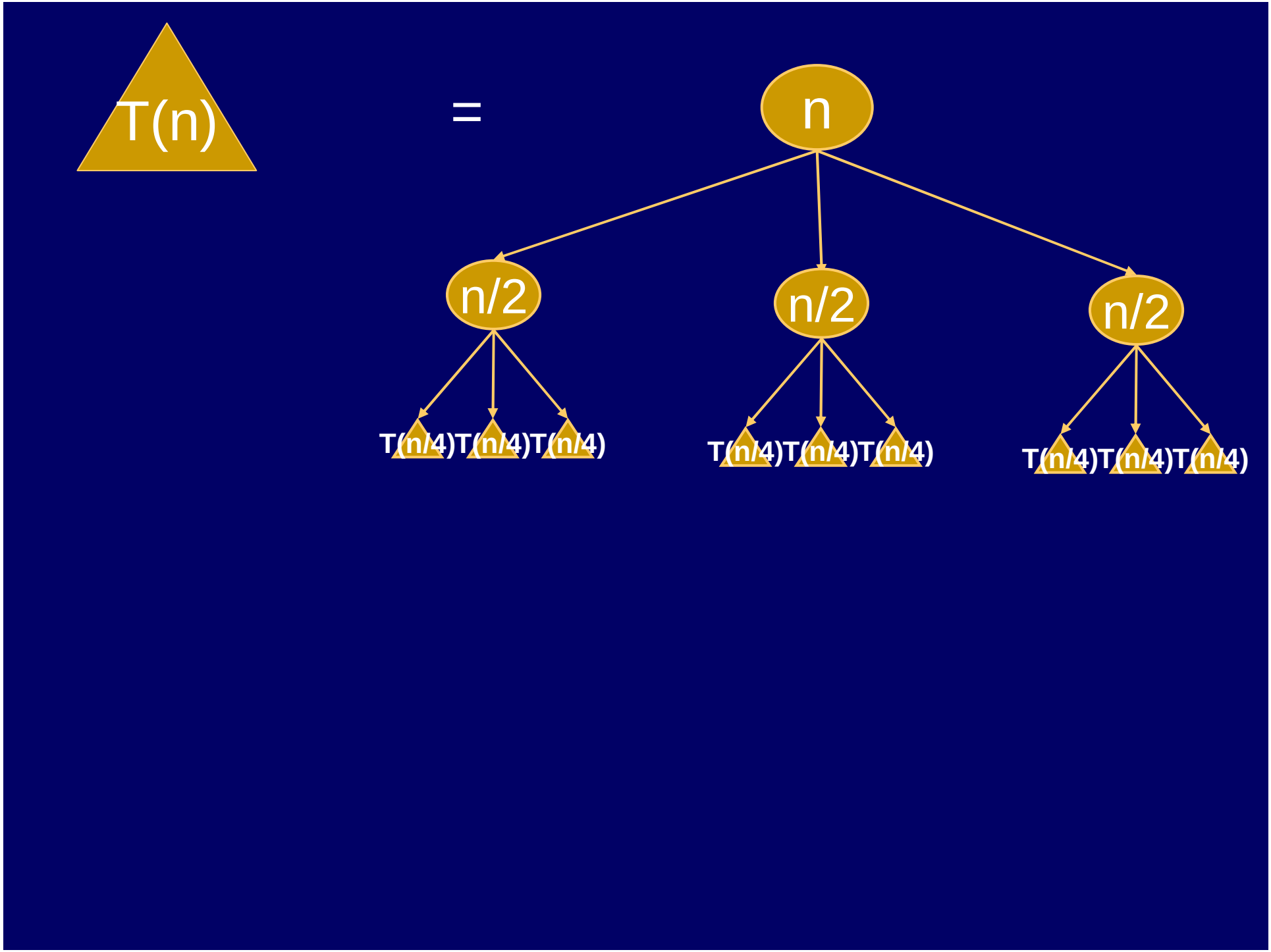
RETURN $e2^n + (MULT(a+b, c+d) - e - f)\, 2^{n/2} + f$

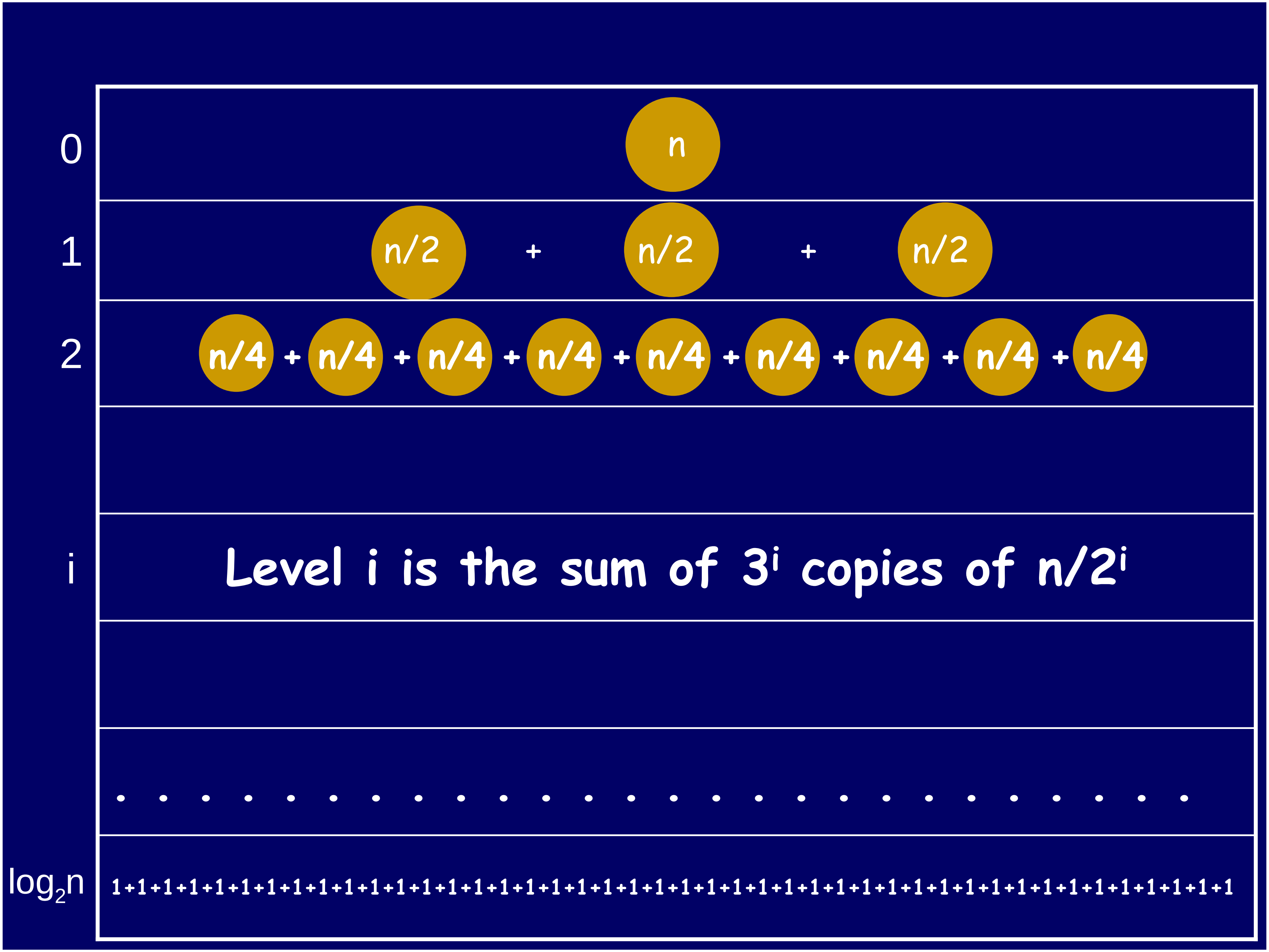
$$T(n) = 3\ T(n/2) + n$$

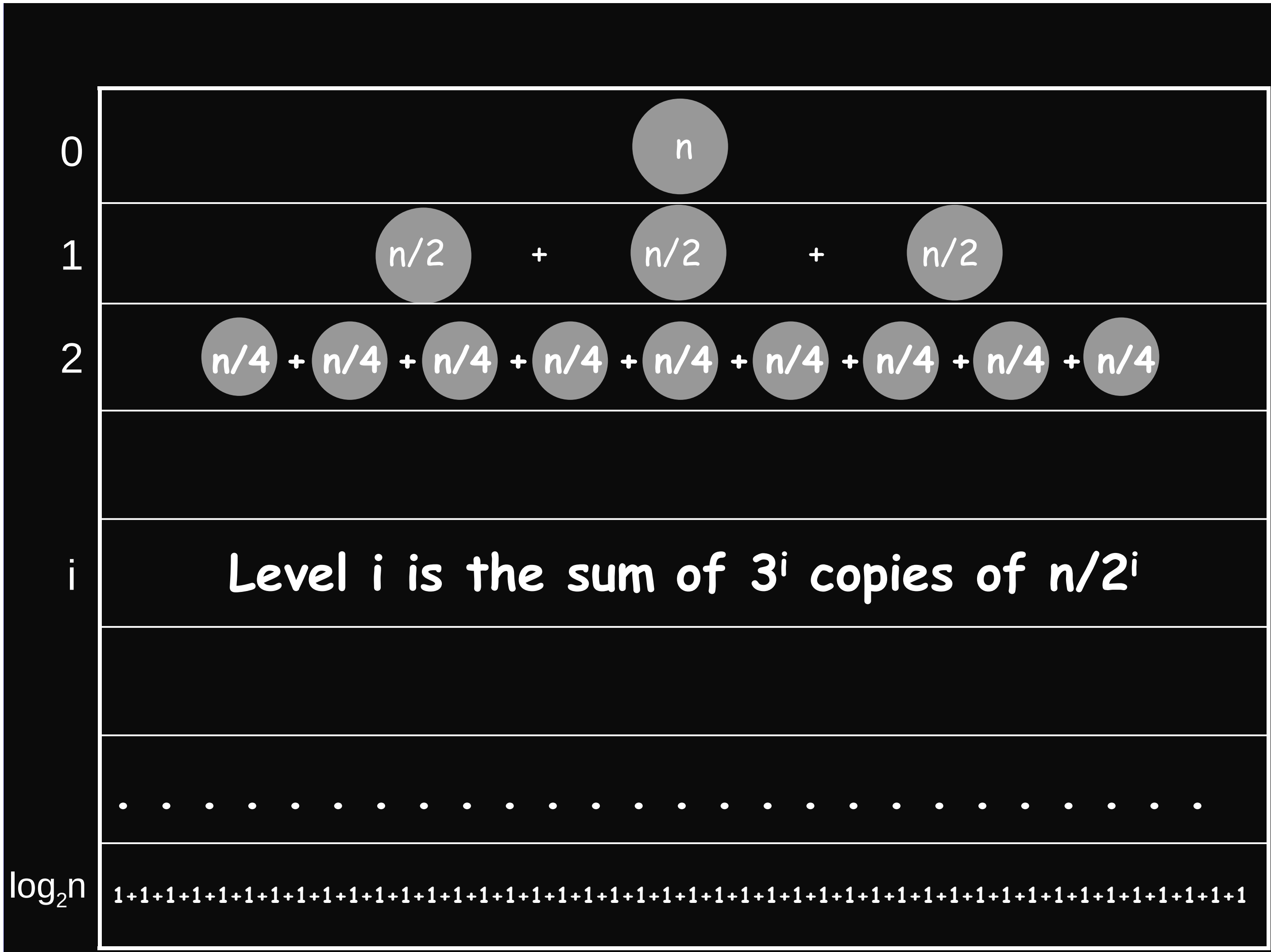
Actually: $T(n) = 2\ T(n/2) + T(n/2 + 1) + kn$


T(n)
=
n
T(n/2)
T(n/2)
T(n/2)


T(n)
=
n
n/2
T(n/2)
T(n/2)
T(n/4)T(n/4)T(n/4)


T(n)
=
n
n/2
n/2
n/2
T(n/4)T(n/4)T(n/4)
T(n/4)T(n/4)T(n/4)
T(n/4)T(n/4)T(n/4)


0
n
1
n/2 + n/2 + n/2
2
n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4 + n/4
i
Level i is the sum of 3^i copies of $n/2^i$
$\log_2 n$
1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1



$= 1n$

$= 3/2n$

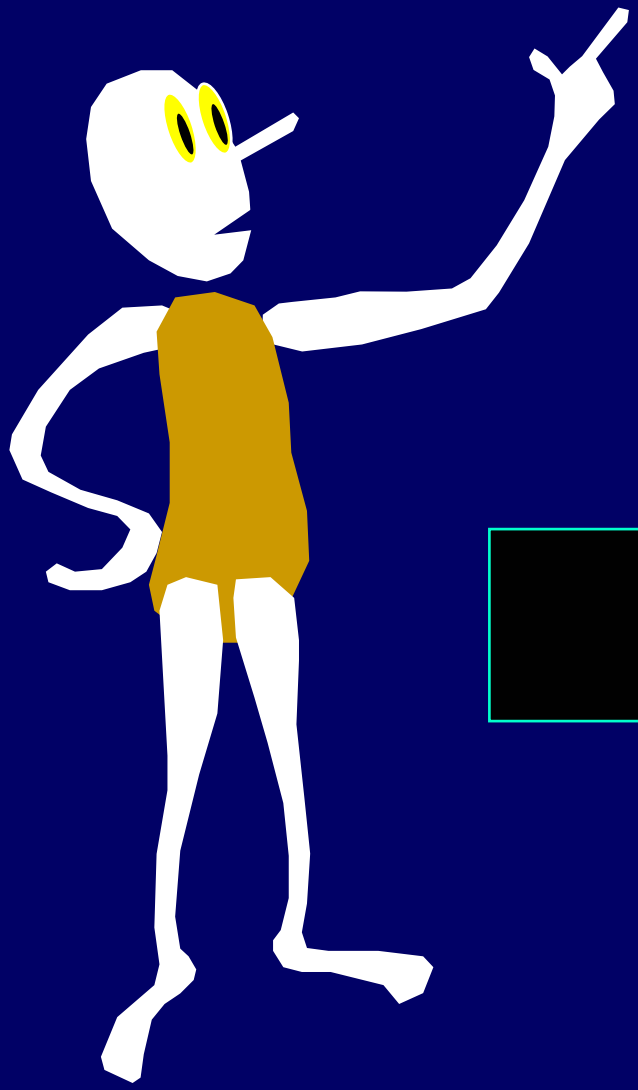
$= 9/4n$

$= (3/2)^i n$

$= \mathbf{(3/2)^{\log_2 n} n}$

$$n(1+3/2+(3/2)^2+ \ldots + (3/2)^{\log_2 n}) = \mathbf{3n^{1.58\ldots} - 2n}$$

$$1 + X^1 + X^2 + X^3 + \ldots + X^{n-1} + X^n = \frac{X^{n+1} - 1}{X - 1}$$



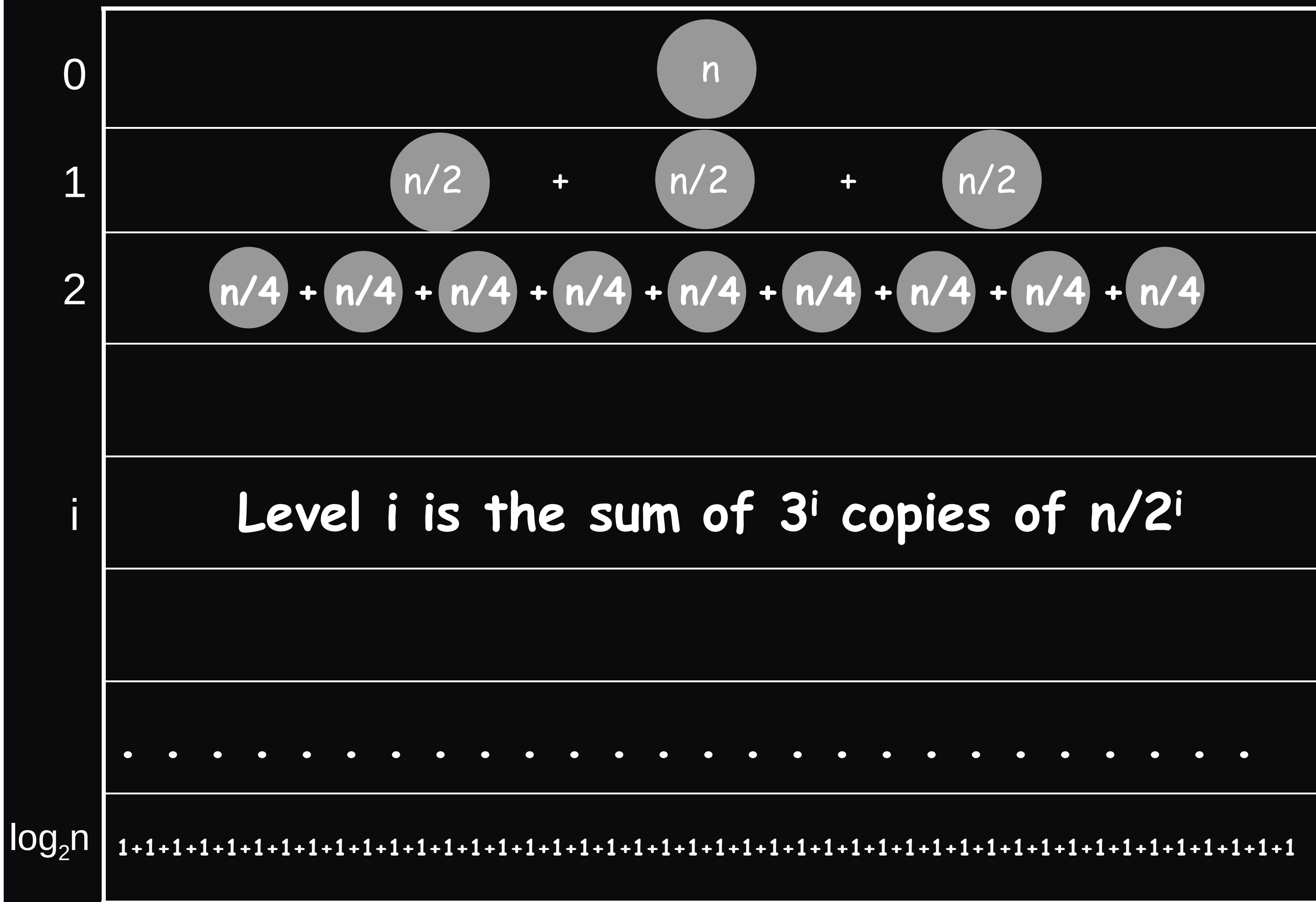
The Geometric Series

Substituting into our formula….

$$1+x+x^2+x^3+\ldots+x^k=\frac{x^{k+1}-1}{x-1}$$

$$x=\frac{3}{2} \qquad k=\log_2 n$$

$$\frac{\left(\frac{3}{2}\right)\frac{3^{\log_2 n}}{2^{\log_2 n}}-1}{\frac{1}{2}}=\frac{(3)3^{\log_2 n}}{n}-2=\frac{(3)3^{\frac{\log_2 3\log_2 n}{\log_2 3}}}{n}-2=\frac{3n^{\log_2 3}}{n}-2$$



$= 1n$

$= 3/2n$

$= 9/4n$

$= (3/2)^i n$

$= \mathbf{(3/2)^{\log_2 n} n}$

$$\mathbf{n\left(\frac{3n^{\log_2 3}}{n} - 2\right)} = 3n^{\log_2 3} - 2n$$

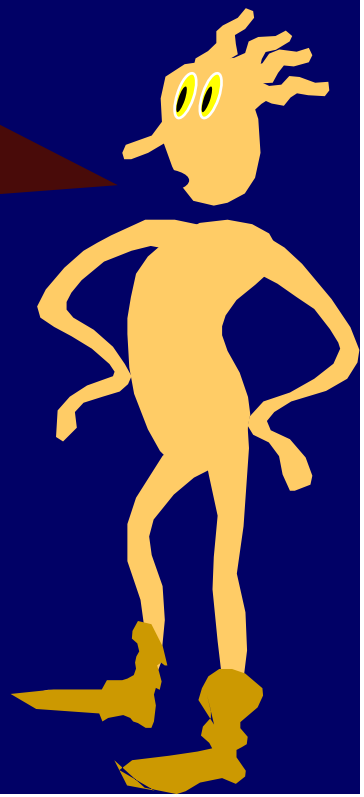
Dramatic improvement for large n

$$3n^{\log_2 3} - 2n = \Theta(n^{\log_2 3}) = \Theta(n^{1.58\ldots})$$

A huge savings over $\Theta(n^2)$ when n gets large.

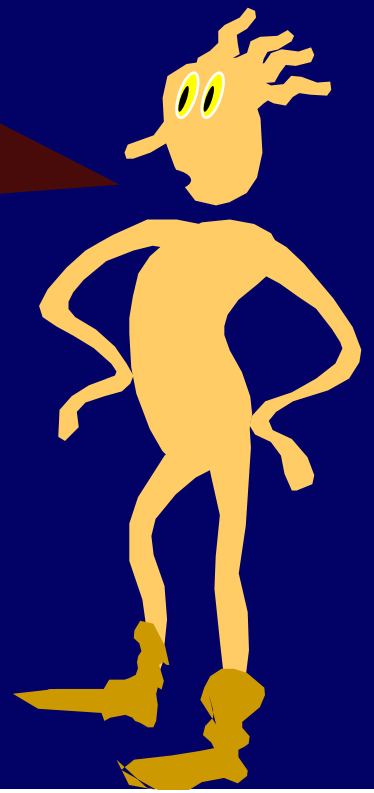
Strange! The Gauss optimization seems to only be worth a 25% speedup. It simply replaces every 4 multiplications with 3. Where did the power come from?

We applied the Gauss optimization RECURSIVELY!



So what is the kissing number of Karatsuba's algorithm?

Not even clear that we kiss once.



Mystery MULT

MULT(X,Y):

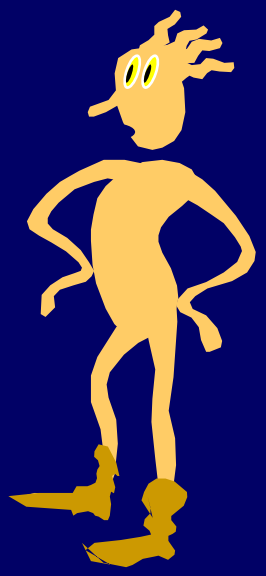
If Y = 0 then RETURN 0

If Y = 1 then RETURN X

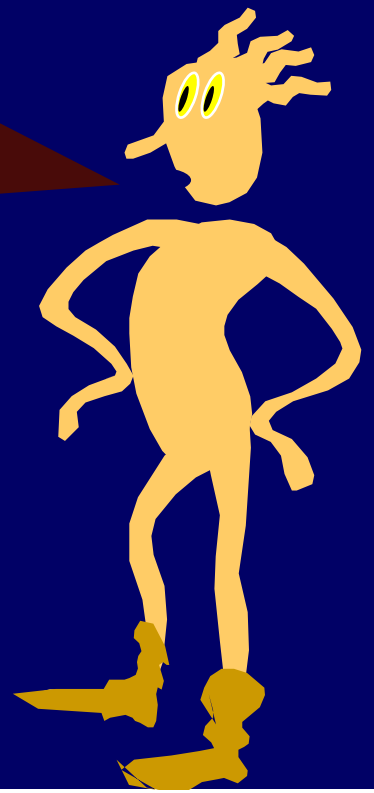
Break Y into c;d, where |d| =1

RETURN 2 [MULT(X,c)] + MULT(X,d)

What is going on? Is this a better way?



That is the Egyptian method sated in recursive language.



Multiplication Algorithms

Kindergarten	$n2^n$
Grade School	n^2
Karatsuba	$n^{1.58\ldots}$
Fastest Known	n logn loglogn

REFERENCES

Karatsuba, A., and Ofman, Y. *Multiplication of multidigit numbers on automata*. Sov. Phys. Dokl. 7 (1962), 595--596.