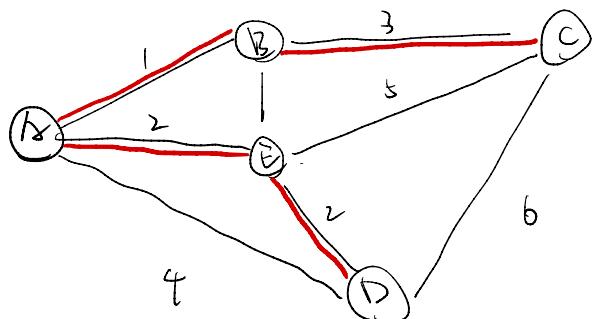


# Lecture-4 02.15.23

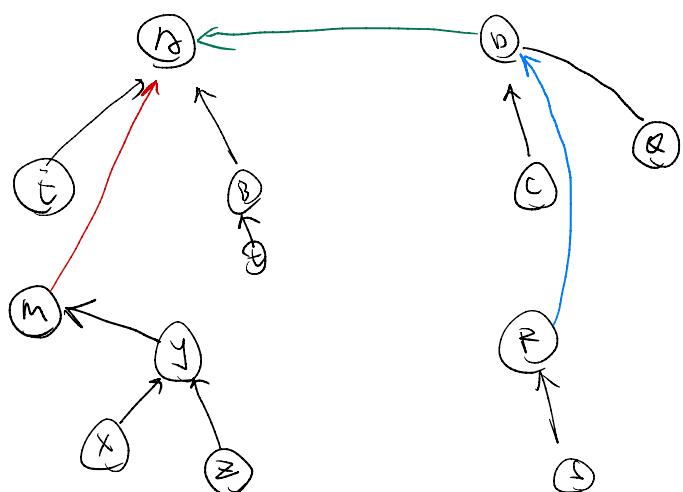
book 23.2 p632



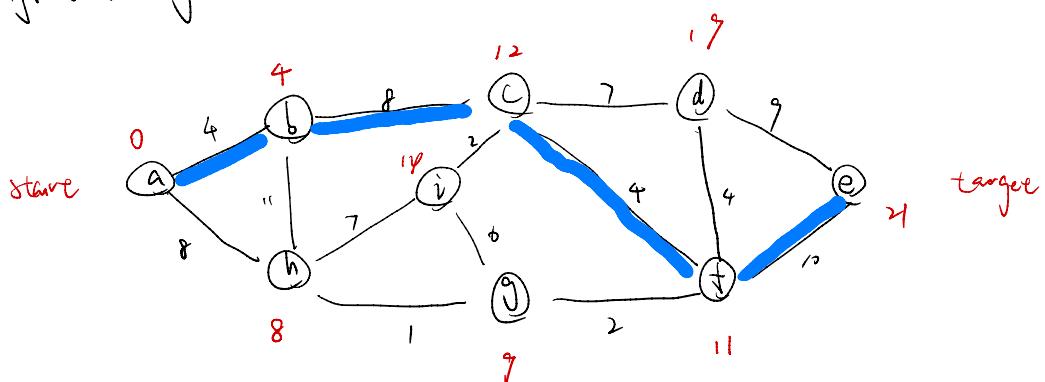
Kruskal Algorithm.

- 1: AB
- 2: ED, AE
- 3: BC

Union Find data structure

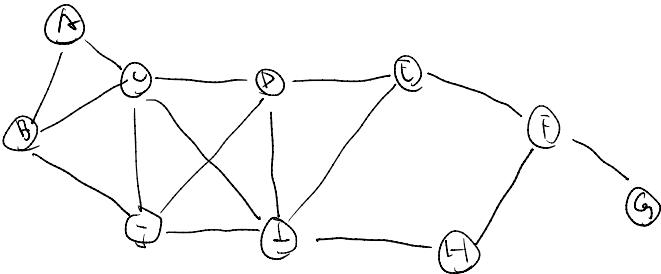


Dijkstra single source shortest path.



a b h g f c d e  
order vertices were visited in the algorithm.

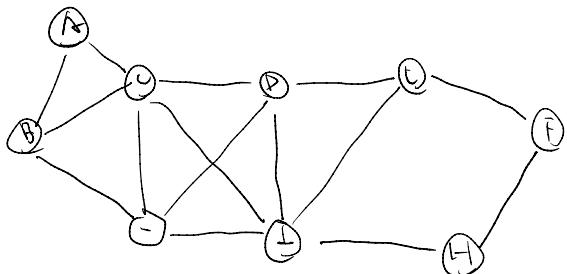
rotating



Which vertex has the minimum degree? G does.

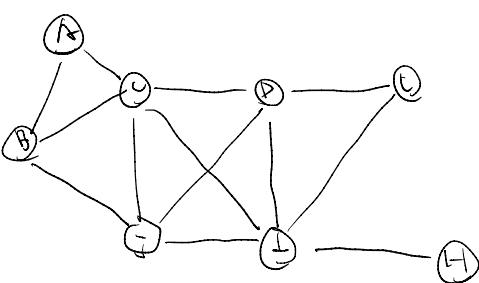
G: 1

Now I'm going to remove G from the graph. at least 3 colors



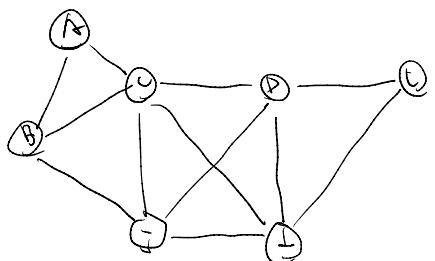
Which vertex has the minimum degree? D. and F.: 2 degree

I'm going to pick up F this time. at least 3 colors.



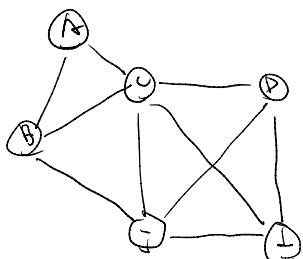
H: 1 degree

at least 3 colors



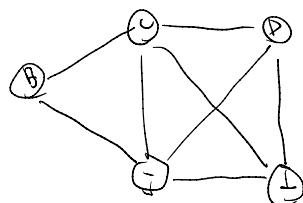
E, D: 2 degree

pick up E. at least 3 color



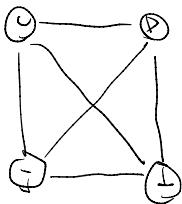
D: 2 degree

at least 3 colors



B: 2 degree

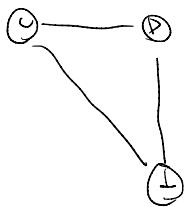
at least 3 colors



C, D, J, I: 3

Pick up J

at least 4 colors



C, D, I: 2

Pick up I.

at least 4 colors.



C, D: 1

Pick up D

at least 4 colors

C

C: 0

at least 4 colors

0	1	2	(3)
C	D	I	J

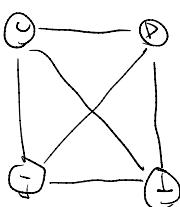
2 2 2 1 2 1

B E F G

complete  
graph

at least 4 colors: J + 3 degree

At the end of this algorithm I ended up with a complete graph, sometimes called a terminal clique. clique is a complete sub graph and terminal means the end.



is a complete graph

complete graph

19	20	25	27	26	(19)
O	P	Q	R	S	T

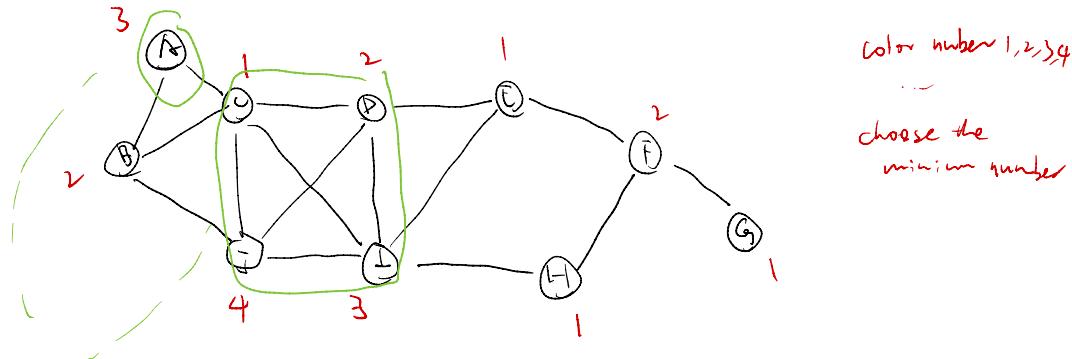
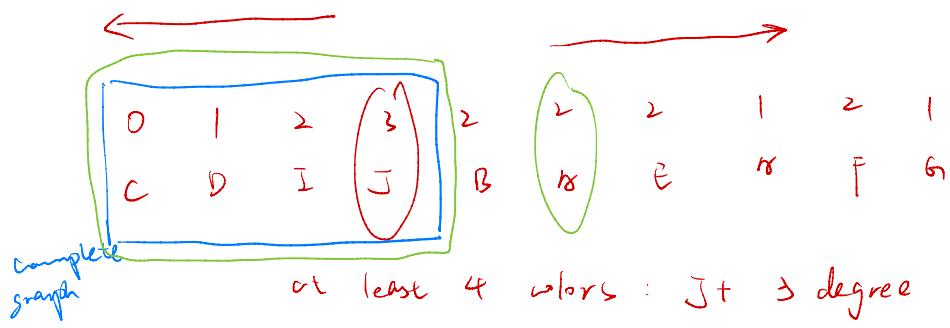
need 30 colors: T + 19 degree

23	24	U	V	W	X	Y	Z
U	V	W	X	Y	Z		

might need 20 colors, actual did it in 19 colors

→ Actual required optimal might be something like 23.

a terminal clique was 20 nearly my upper bound was 20. in the  
be 11 3

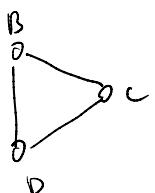
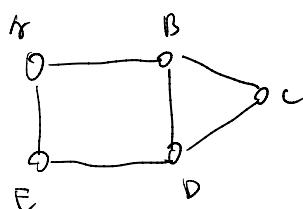


all of these in here needed one more than what they had when we deleted them.

SINCE I know C, D, J, I all have 3 degrees  $\Rightarrow$  a complete graph  
Therefore, I know it at least need 4 colors (low bound)

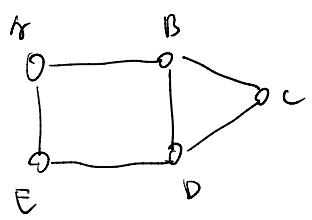
And also find 4 is right part upper bound.

However, the last terminal graph might not the largest complete graph, just limit max and min.

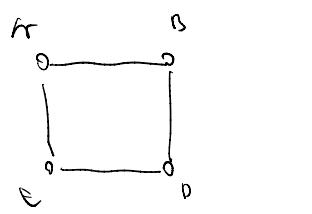


biggest complete graph have

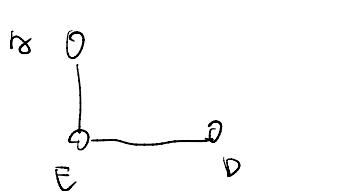
start from c.



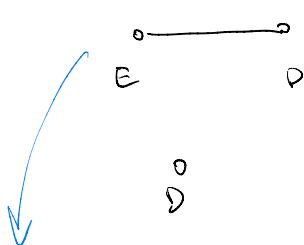
$C$  degree: 2  
pick up



$A, E, D, B$  degree: 2  
pick up  $B$

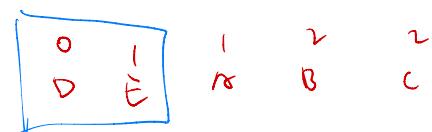


$A, D$  degree: 1  
pick up  $A$



$E, D$  degree: 1  
pick up  $E$

$D$  degree: 0



complete graph lower bound = 2 colors

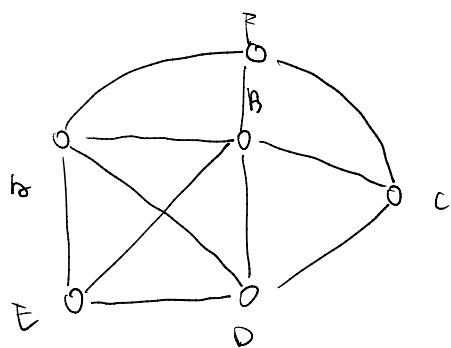
upper bound = 3 colors

I can do it in 3. I know I require 2

是经济找一个一生最大的圆. 用这个圆来覆盖所有的圆

这个圆叫做上下面圆.

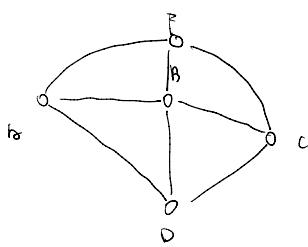
这个圆叫做上下面圆.



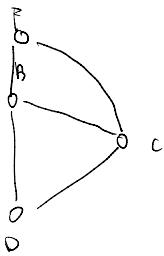
3

more from  $\Sigma$  first

$\Sigma$

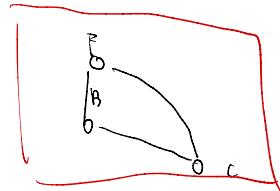


$\frac{3}{A}$   $\frac{3}{E}$

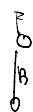


$\frac{2}{D}$   $\frac{3}{A}$   $\frac{1}{E}$

complete  
graph



$\frac{2}{C}$   $\frac{2}{D}$   $\frac{3}{A}$   $\frac{1}{E}$



$\frac{1}{F}$   $\frac{2}{C}$   $\frac{2}{D}$   $\frac{3}{A}$   $\frac{1}{E}$



0	1	2	2	<u><math>\frac{3}{A}</math></u> $\frac{1}{E}$
B	F	C	D	

I might need four colors will accomplished.

$\frac{3}{}$  minimum required.

Discrete log problem

Diffie Hellman key exchange

Finding  $x$  is hard

$$2^x \% 11 = 10$$

$$(2^x)^y = (2^y)^x \quad (2^x)^y \% m = (2^y)^x \% m$$

$$(2^x \% m)^y \% m = (2^y \% m)^x \% m$$

$\geq$  base  $11$  modulus

A

B

choose random:  $a=3$

b choose 4

$$\begin{array}{ccc} & 8 & \\ \xrightarrow{\hspace{2cm}} & & \\ 2^a \% 11 = 8 & & \\ & \swarrow 5 & \\ (2^b \% 11)^a \% 11 & & \text{it doesn't know } b \text{ how is} \\ & & \text{but it knows } (2^b \% 11) = 5 & (2^a \% 11)^b \% 11 \end{array}$$

$$5^a \% 11$$

$$5^b \% 11 = 4 \quad 8^4 \% 11 = 4$$

$\text{mod } 7$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

$2 \times 5 \% 7 = 3$

$2 \times 6 \% 7 = 5$

$$(3 \times \frac{1}{3}) = 1$$

$$(3 \times \frac{1}{3}) \% 7 = 1$$

$$(3 \times a) \% 7 = 1 \Rightarrow a = \frac{1}{3}$$

$$(3 \times 5) \% 7 = 1 = (3 \times \frac{1}{3}) \% 7$$

$$\frac{1}{3} \% 7 = 5$$

$$\frac{1}{3} + \frac{1}{3} + \frac{1}{3} = 1$$

$$(5 + 5 + 5) \% 7 = 1$$

$$(6 \cdot \frac{1}{3}) \% 7 = 2$$

$$(6 \cdot 5) \% 7 = 2$$

$$(\frac{1}{2} \% 7) = ?$$

$$(2 \% 7) = 2$$

$$((2 \times \frac{1}{2}) \% 7) = 1 \% 7 = 1$$

$$(2 \times a) \% 7 = (2 \times 4) \% 7 = 1$$

$$\frac{1}{2} \% 7 = 4$$

$$\frac{1}{6} \% 7 = ?$$

$$(6 \times \frac{1}{6}) \% 7 = 1 \% 7 = 1$$

$$(6 \times 6) \% 7 = 1$$

$$\frac{1}{6} \% 7 = 6$$

$$\frac{1}{4} \% 7 = 2$$

$$\begin{aligned}\frac{1}{3} \% 7 &= 5 \\ \frac{1}{4} \% 7 &= 4 \\ \frac{1}{6} \% 7 &= 1 \\ \frac{1}{4} \% 7 &= 2\end{aligned}\quad \begin{aligned}\frac{1}{4} + \frac{1}{4} + \frac{1}{2} &= 1 \\ (2 + 2 + 4) \% 7 &= 1 \\ (\frac{1}{3} + \frac{1}{2} + \frac{1}{6}) \% 7 &= (5 + 4 + 6) \% 7 = 1\end{aligned}$$

You can do all of these operations over just integer. It's quite powerful. It's not only for cryptography, it gets used in error correction codes as well and other things as an example in the following:

Ferd Sullivan Erasure Coding which is sometimes used in raid arrays and other types of things basically says that you have three data items. You know you're going to lose some things. So you want to add some redundancy in order to lose things. So say you have three data items you want to lose be able to lose how many? For example, you don't want to be able to lose two of them. Then you need five altogether.

Store on the first disk would be  $x$  which would sort of the second disk would be  $y$ , which store on the third disk would be  $z$ ...

Suppose

<del>lose</del>	<del>1st disk</del>	$x$	0-6 bits
	2nd disk	$y$	0-6 bits
<del>lose</del>	<del>3rd disk</del>	$z$	0-6 bits
	4th disk	$x+y+z$	$(x+y+z) \% 7$ 0-6 bits
	5th disk	$x+2y+3z$	$(x+2y+3z) \% 7$ 0-6 bits
<del>lose</del>	<del>6th disk</del>	$x+4y+9z$	$(x+4y+9z) \% 7$ 0-6 bits

Let's say you lose 1st, 3rd, 6th. Now you have  $y, (x+2y+3z) \% 7$ .

$$(x+4y+9z) \% 7$$

Galois

$$\frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

$$2 \cdot \frac{1}{3} = \frac{2}{3}$$

$$(5+5) \% 7 = 3$$

$$(2 \times 5) \% 7 = 3$$

$$(\frac{1}{2} + \frac{1}{6}) \% 7 = (4+6) \% 7 = 3$$

$$(\frac{1}{4} + \frac{1}{4} + \frac{1}{6}) \% 7 = (2+2+6) \% 7 = 3$$

$$(\frac{1}{2} \cdot 6) \% 7 = 3$$

$$(4 \cdot 6) \% 7 = 3$$

$$(\frac{1}{3} \cdot 6) \% 8 = 3$$

$$(\frac{1}{5} \cdot 6) \% 8 = 5$$

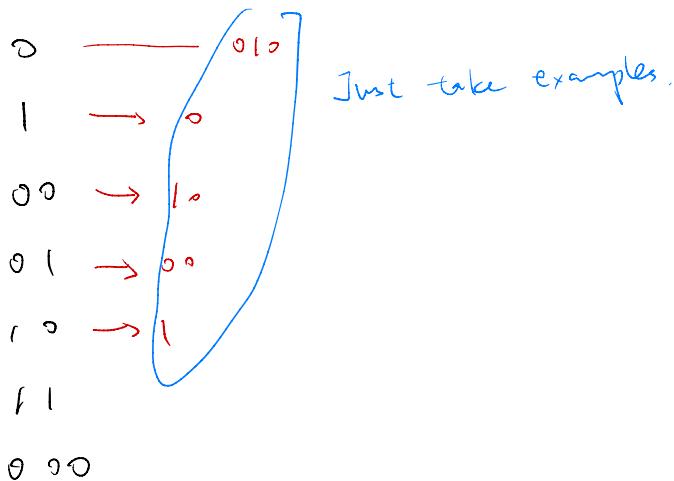
$$\frac{1}{7} \% 8 = 7$$

$$\frac{1}{4} \% 8 = \text{Don't Exist}$$

mod % 8	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

$$(4 \times \frac{1}{4}) \% 8 = 1 \% 8 = 1$$

$\frac{1}{4} \% 8$   
 relatively prime



0 0 1

0 1 0 → 0 1

lossless compression algorithm takes 010

and it makes it a zero one. That's pretty cool

0 1 1

I saved myself a bit.

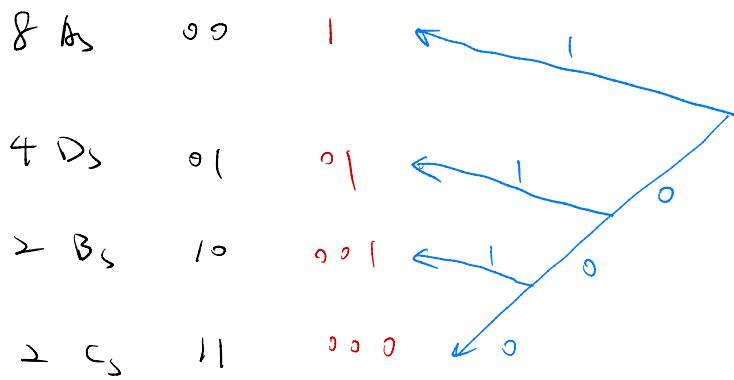
1 0 0

1 0 1

1 1 0

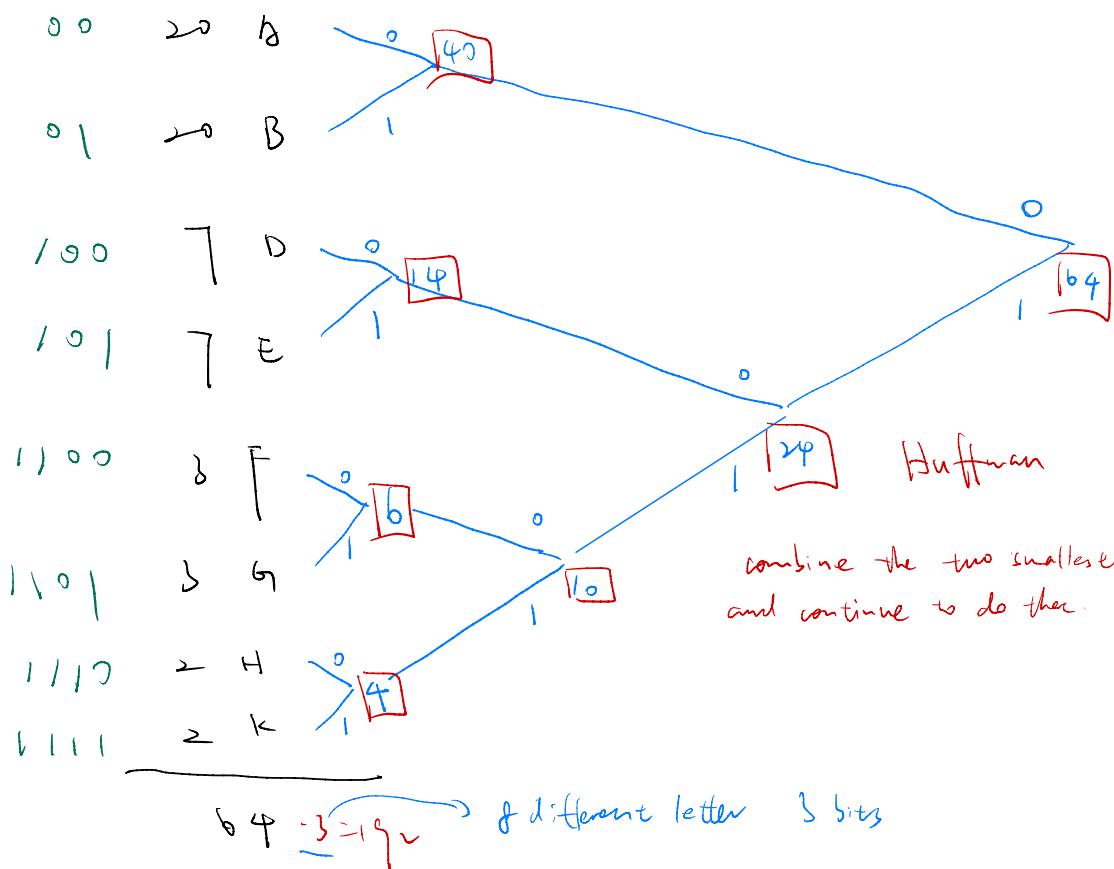
1 1 1

$\begin{matrix} 00 & 10 & 00 & 10 & 00 & 11 & 00 & 11 & 000 & 01 & 00 & 01 \\ A & B & B & B & C & C & C & C & D & D & D & D \end{matrix}$   
 $1 \quad 000 \quad 1 \quad 000 \quad 1 \quad 001 \quad 1 \quad 001 \quad 1 \quad 01 \quad 1 \quad 01 \quad 1 \quad 01$   
 $2 \times 16 = 32 \text{ bits}$   
 $1 \quad 01 \quad 1 \quad 01 = 28 \text{ bits}$



This is kind of deciding tree which value or which letter I got through.

It's called prefix encoding when you do that.



$$\frac{20 \cdot 2 + 20 \cdot 2}{80} + \frac{7 \cdot 3 + 7 \cdot 3}{42} + \frac{10 \cdot 4}{40} = 16.2$$