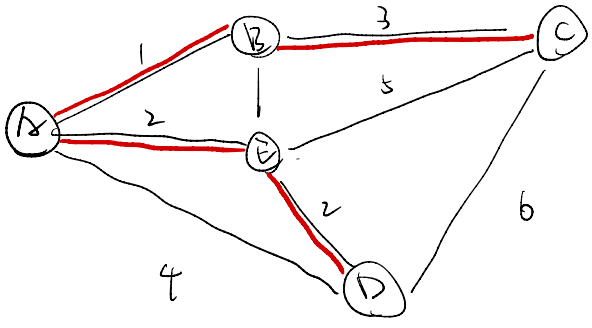
Lecture_4 02.15.23

book 23.2 p 632



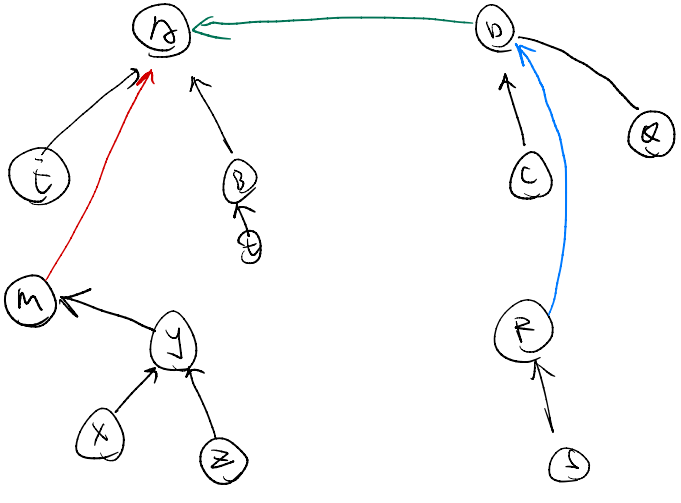
kruskal Algorithm

1: AB

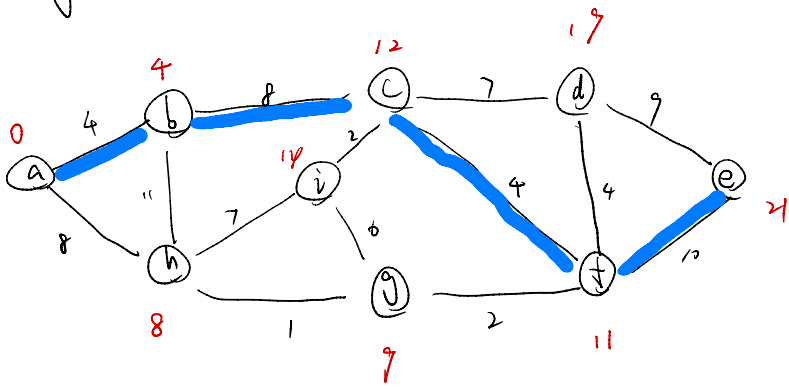
2: ED, AE

3: BC

Union Find data structure



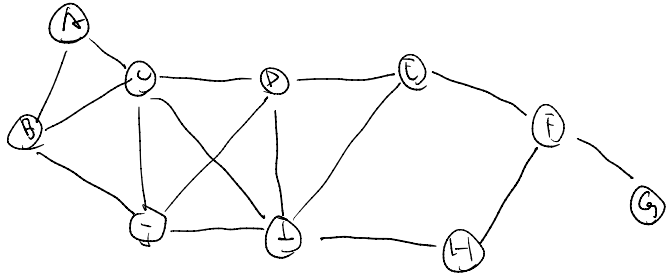
Dijkstra single source shortest path.



a b h g f c d e

order vertices were visited in the algorithm.

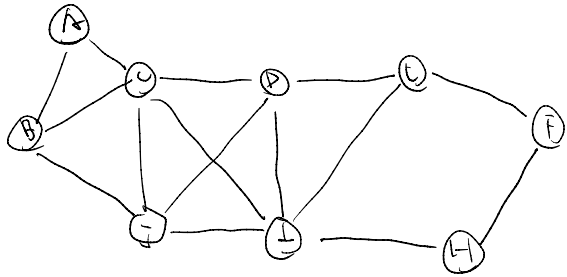
coloring



Which vertex has the minimum degree? G does.

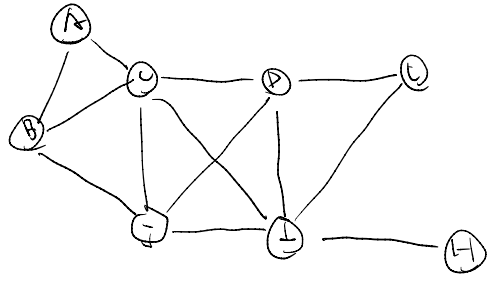
G: 1

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



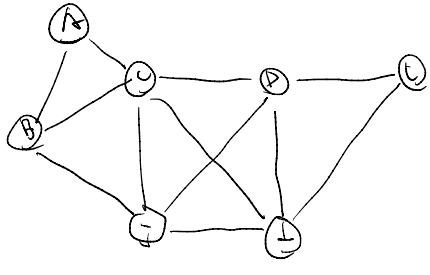
Which vertex has the minimum degree? G. and F. : 2 degree

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



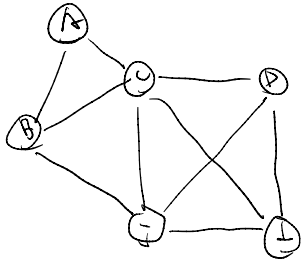
H: 1 degree

at least 3 color

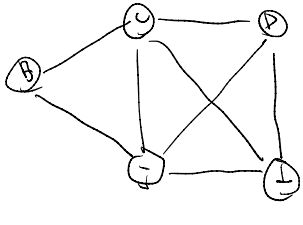


E. G: 2 degree

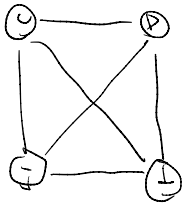
pick up E. at least 3 color



G: 2 degree . at leas 3 color



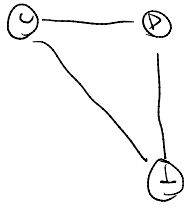
B: 2 degree at lease 3 colors



C, D, J, I : 3

pick up J

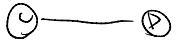
at least 4 colors



C, D, I : 2

pick up I.

at least 4 colors.



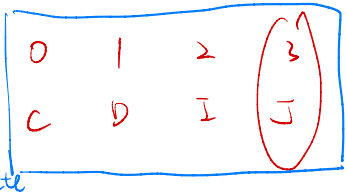
C, D : 1

pick up D

at least 4 colors

C : 0

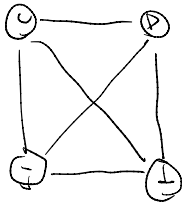
at least 4 colors



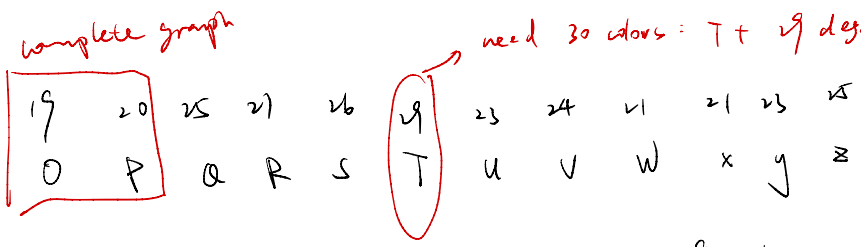
complete graph

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

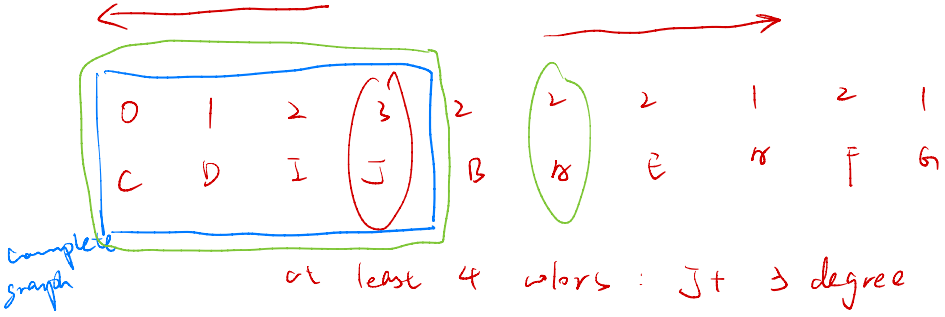


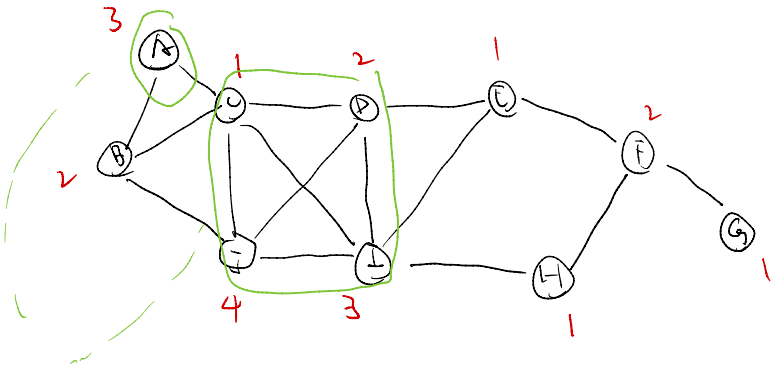
might need 30 colors. actual did it in 19 colors

→ Actual required optimal might be something like 23

a terminal clique was 20 meaning my upper bound was 20. in the

be 11 3





color number 1, 2, 3, 4

choose the minimum number

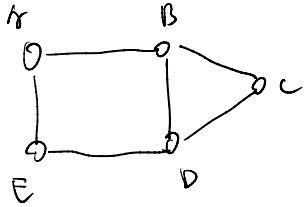
all of these in here needed one more than what they had when we deleted them.

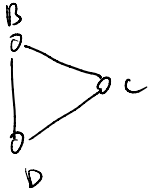
SINCE I know C, D, J, I all have 3 degrees ⇒ a complete graph

Therefore, I know it at least need 4 colors (low bound)

And also Find 4 is right part upper bound.

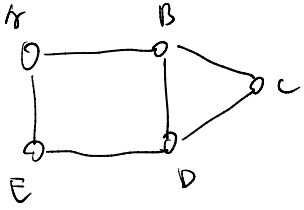
However, the last terminal graph might not the largest complete graph. just limit Max and min.





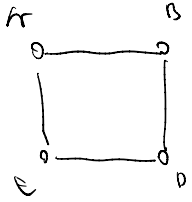
biggest complete graph here.

start from C.



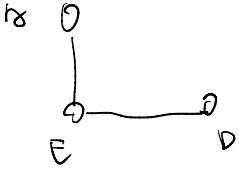
C degree: 2

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

0	1	1	2	2
D	E	A	B	C

complete graph

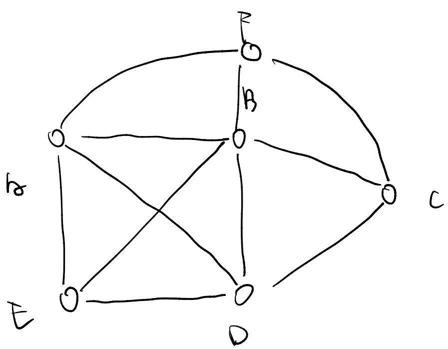
lower bound = 2 colors

upper bound = 3 colors

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

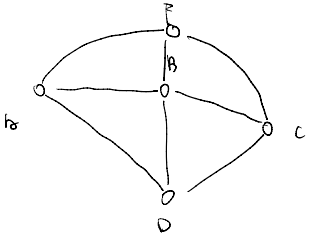
最终的找一个不一定是最大的全图（说到底一个不同，是局部完全图）也不同，所以会有上下限范围。

→ 需要进行一些其他的分析

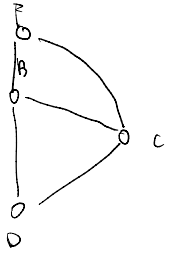


3
E

move from E first

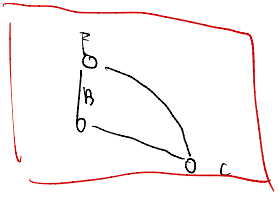


3 3
A E

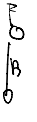


2 3 1
D A E

complete graph



2 2 3 1
C D A E



1 2 2 3 1
F C D A E

0 1 2 2 3 1
B F C D A E

I might need four colors will accomplished.

3 minimum required.

Discrete log Problem

finding x is hard

Diffie Hellman key exchange

$2^x \% 11 = 10$

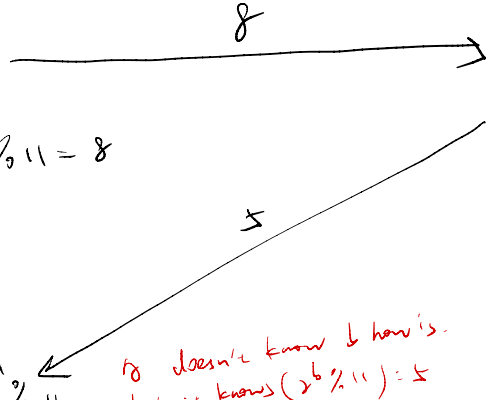
$(2^x)^y = (2^y)^x \qquad (2^x)^y \% m = (2^y)^x \% m$

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

2 base 11 modules

A B

choose random : a=3 b choose 4



A doesn't know b how is. but it knows $(2^b \% 11) = 5$

$(2^b \% 11)^a \% 11$

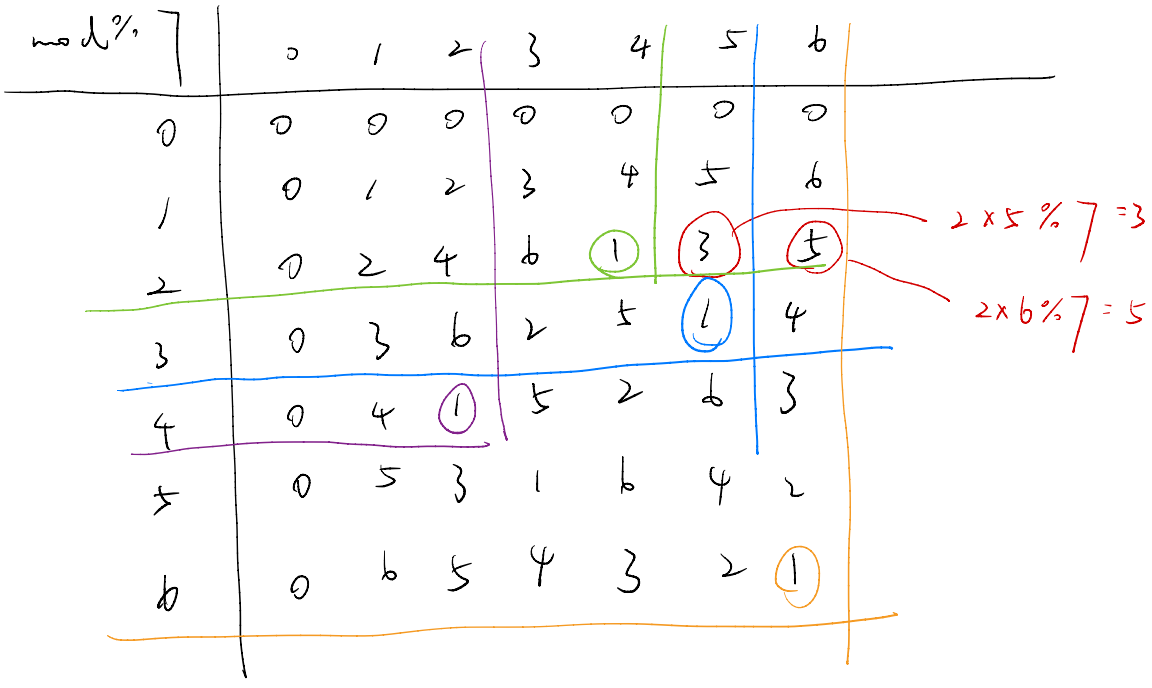
$5^a \% 11$

$5^3 \% 11 = 4$

$(2^a \% 11)^b \% 11$

$8^b \% 11$

$8^4 \% 11 = 4$



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

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

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

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

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

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

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

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

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

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

$(2 \% 7) = 2$

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

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

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

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

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

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

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

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

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

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

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

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

$\frac{1}{4} + \frac{1}{4} + \frac{1}{2} = 1$

$(2 + 2 + 4) \% 7 = 1$

$(\frac{1}{3} + \frac{1}{2} + \frac{1}{6}) \% 7 = (5 + 4 + 6) \% 7 = 1$

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:

Read 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
lose	~~1st disk~~	x	0-6 bit
	2st disk	y	0-6 bit
lose	~~3rd disk~~	z	0-6 bit
	4th disk	x+y+z	(x+y+z)%7 0-6 bits
	5th disk	x+2y+3z	(x+2y+3z)%7 0-6 bits
lose	~~6th disk~~	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

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

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

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

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

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

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

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

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

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

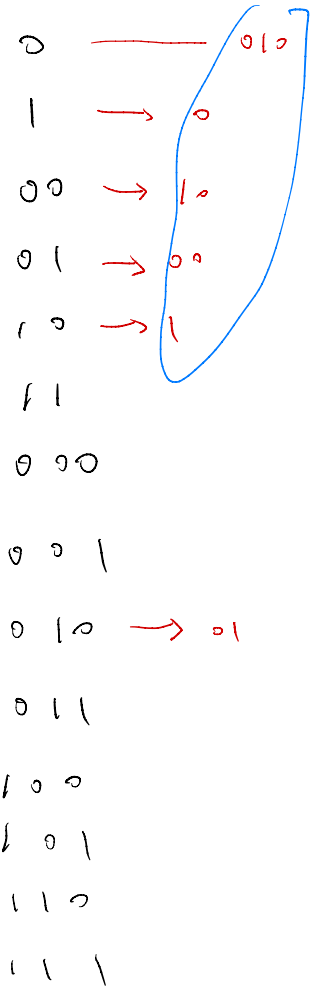
$\frac{1}{4}\%8 =$ don't exist.

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

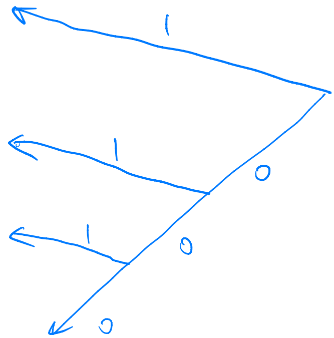


Just take examples.

lossless compression algorithm takes 010 and it makes it a zero one. That's pretty cool I saved myself a bit.

00	10	00	10	00	11	00	11	00	01	00	01	00	01	00	01	= 2×16 = 32 bit
A	B	A	B	A	C	A	C	A	D	A	D	A	D	A	D	
1	000	1	000	1	001	1	001	1	01	1	01	1	01	1	01	= 28 bit

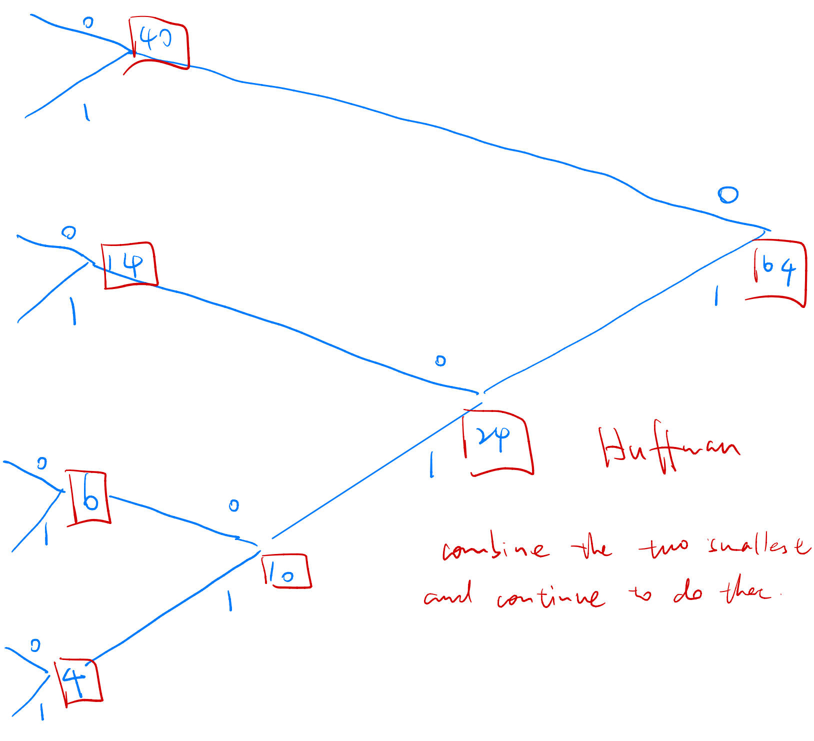
8 As	00	1
4 Ds	01	01
2 Bs	10	001
2 Cs	11	000



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

It's called prefix encoding when you do that.

00	20	A
01	20	B
100	7	D
101	7	E
1100	3	F
1101	3	G
1110	2	H
1111	2	K



Huffman

combine the two smallest and continue to do that.

64 · 3 = 192 → 8 different letter 3 bits

$$\underbrace{20 \cdot 2 + 20 \cdot 2}_{80} + \underbrace{7 \cdot 3 + 7 \cdot 3}_{42} + \underbrace{10 \cdot 4}_{40} = 162$$