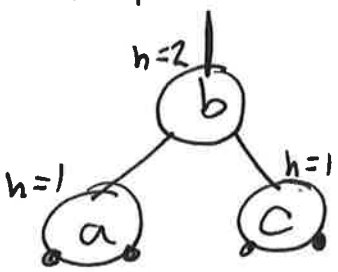
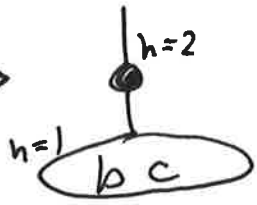
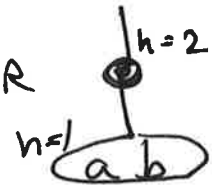
Black holes always start when data is deleted from a leaf n=1 node with parent n=1 & bro/sis n=1:



depending on a or c deleted →



OR



(BH's only purpose is to keep subtree's height unchanged.)

SO FAR BHs always have one child with n=2
let's hope (but verify) that's all we ever have to see.

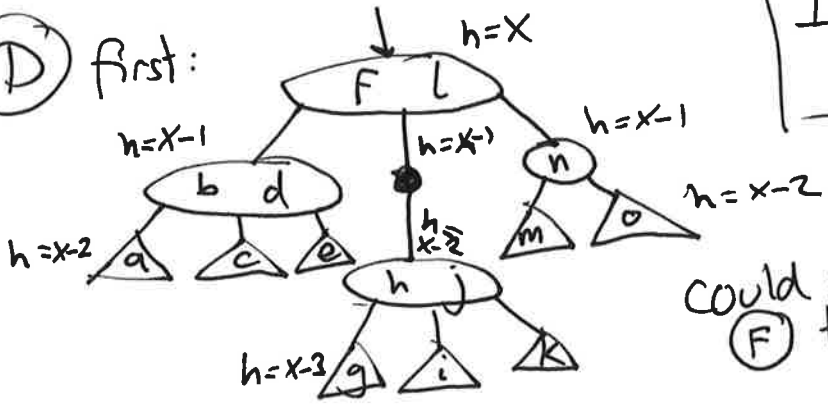
What happens depends on the parent node
(unseen at h=3)

two cases: (A) it's an n=2 node
(B) it's an n=1 node

(A) first, will only look at case of BH in the middle, left & right left to you.
three cases for that: (C) both bro/sis have n=2
(D) one has n=1 other n=2
(E) both n=1

I will only look at (D) & (E), (C) is left to you

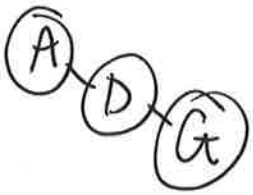
(A) (D) first:



only x-3 can be zero

could say
(F) treat it as 7 data items
b, d, f, h, j, l, n
& 8 subtrees a, c, e, g, i, k, m, o

OR (G) 5 data items b, d, f, l, n
& 6 subtrees a, c, e, (g-k), m, o



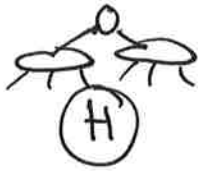
is smallest so try A first

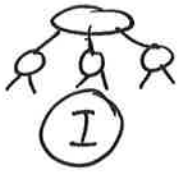
everything we've got, in order:

data b, d, f, L, n

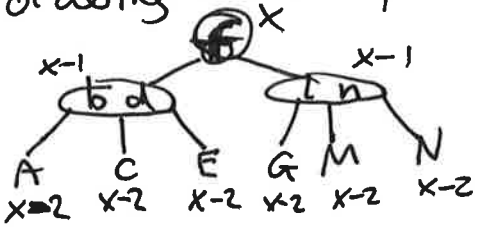
subtrees a, c, e, (g-k), m, o

subtr. heights x-2 x-2 x-2 x-2 x-2 x-2

Could reassemble as 

or 

ordering would require



it works! overall height same
all balanced

so don't even waste time looking at (I)

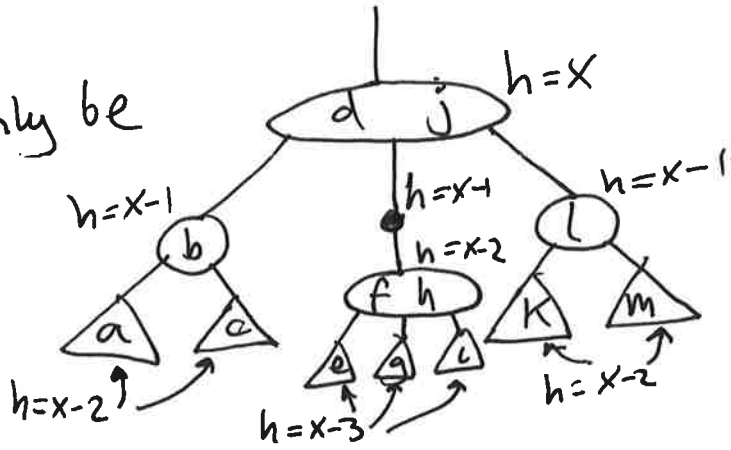
Case (G) is complete

∴ no need to look at (F) either.

Case (D) is complete

And the black hole has gone.

A E can only be



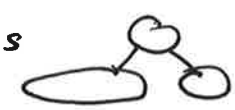
could consider to be (J) 6 data items: b, d, f, h, j, l
& 7 subtrees: a, c, e, g, i, k, m

or (K) 4 data items: b, d, j, l
& 5 subtrees: a, c, (e-i), k, m

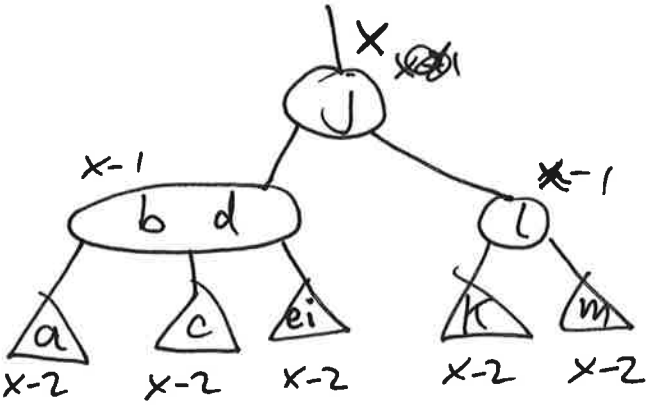
A E K smallest so check first

everything we've got, in order

data	b,	d,	j,	l	
subtrees	a,	c,	(e-i),	k,	m
st heights	x-2	x-2	x-2	x-2	x-2

can only be arranged as  or 

and both work, check heights



overall height unchanged
& black hole has gone.
case (E) solved ∴ case (A) solved.

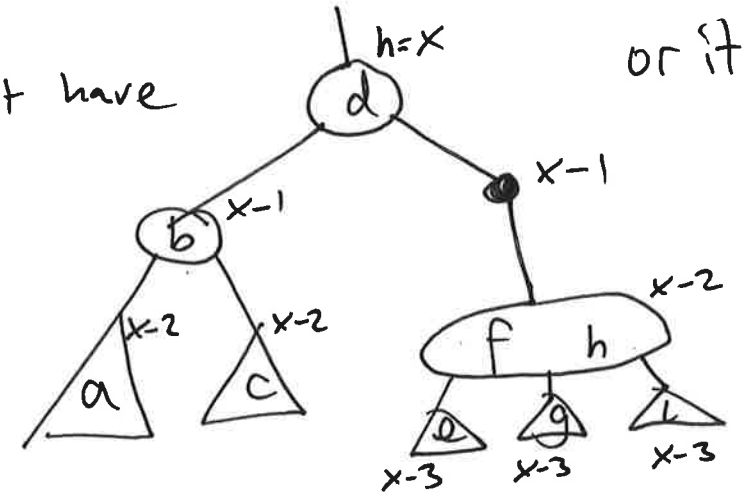
(B) parent of BH has $h = 1$.

two cases (L) bro/sis has $n=1$

(M) bro/sis has $n=2$

I will only look at (L) leave (M) to you.

(B)(L) must have



or its mirror image

could consider to be (N) 4 data items: b, d, f, h
& 5 subtrees: a, c, e, g, i

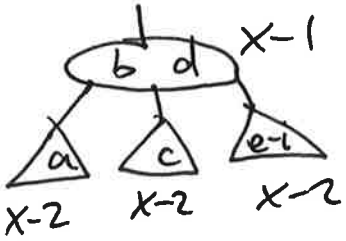
or (O) 2 data items: b, d
& 3 subtrees: a, c, (e-i)

(B)(L)(O) smallest, try it first.

What we've got, in order

data	b,	d	
subtrees	a,	c,	(e-i)
hts	x-2	x-2	x-2

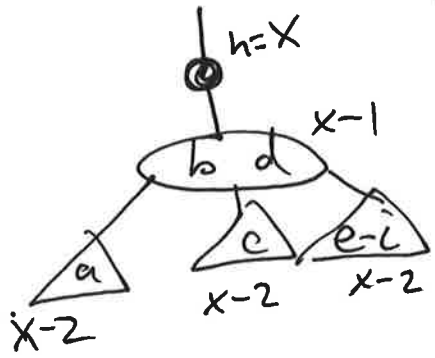
easy:



← subtree height has changed. Bad.

continued

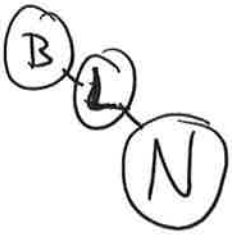
subtree height must not change ∴ we really have



that is a valid solution,

but maybe we can do better, get rid of BH

so try the other option, (N)



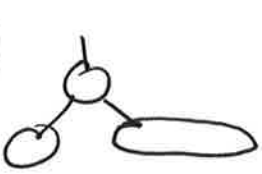
What we've got, in order:

data: b, d, f, h

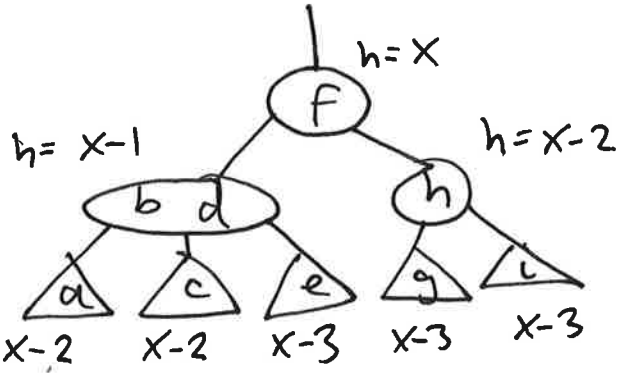
subtr: a, c, e, g, i

hts: x-2 x-2 x-3 x-3 x-3

4 data items must be 

or 

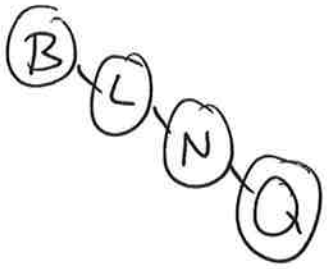
try (P): populate & check heights



not balanced

so not a solution

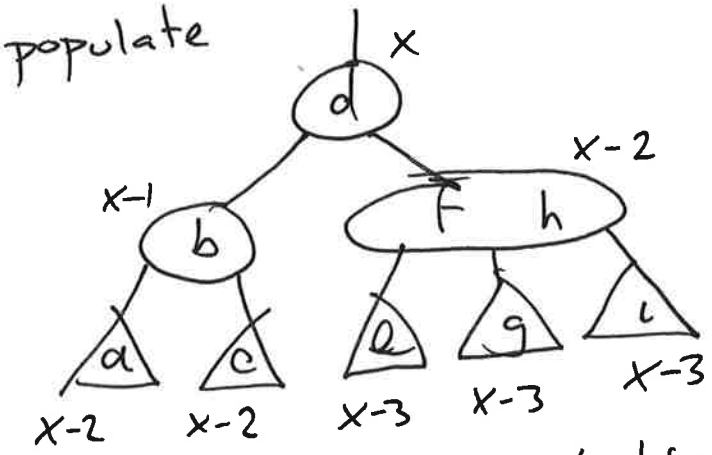
move on to (Q)



got b, d, f, h

~~got~~ trees a, c, e, g, i

hts x-2 x-2 x-3 x-3 x-3



not balanced so no solution

∴ case (B) can only be solved via option (O)

the black hole continues up the tree.

But at least we know black holes can only have an n=2 child.

So a whole lot of possibilities are not possible at all ∴ can be ignored.