from A :	A	B	C	D	E	F	G
distance	0	5	5	8	9	4	9
parent	A	F	B	C	D	A	F

index	0	1	2	3	4	5	6	7	8	9
H at start	10	9	8	7	6	5	4	3	2	1
after BUP(1)	9	10	8	7	6	5	4	3	2	1
after BUP(2)	8	10	9	7	6	5	4	3	2	1
after BUP(3)	7	8	9	10	6	5	4	3	2	1
after BUP(4)	6	7	9	10	8	5	4	3	2	1
after BUP(5)	5	7	6	10	8	9	4	3	2	1
after BUP(6)	4	7	5	10	8	9	6	3	2	1
after BUP(7)	3	4	5	7	8	9	6	10	2	1
after BUP(8)	2	3	5	4	8	9	6	10	7	1
after BUP(9)	1	2	5	4	3	9	6	10	7	8

index	0	1	2	3	4	5	6	7	8	9
H at start	10	9	8	7	6	5	4	3	2	1
after TDN(4)	10	9	8	7	1	5	4	3	2	6
after TDN(3)	10	9	8	2	1	5	4	3	7	6
after TDN(2)	10	9	4	2	1	5	8	3	7	6
after TDN(1)	10	1	4	2	6	5	8	3	7	9
after TDN(0)	1	2	4	3	6	5	8	10	7	9

Indices: root 0, last element $n-1 = 2^t - 2$. $2^{t-1} - 2$ down to 0.

$t = 1$: wc kc $= 2 = 2 * 1$

$t = 2$: wc kc $= 2 + 2 + 4 = 2 * (1 + 1 + 2)$

$t = 3$: wc kc $= 2 + 2 + 2 + 2 + 4 + 4 + 3 = 2 * (1 + 1 + 1 + 1 + 2 + 2 + 3)$

$2 * \sum_{j=1}^{t-1} j 2^{(t-1)-j}$.

It can be shown that this is in $\Theta(n)$.

So building a heap with n keys takes at most $\Theta(n)$ time.