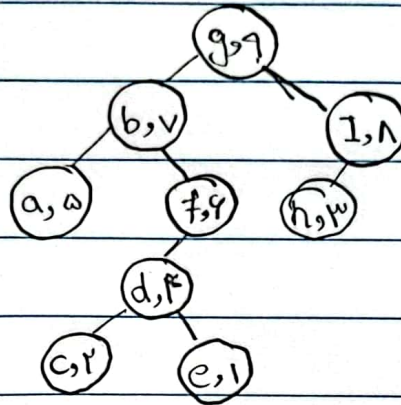


a	b	c	d	e	f	g	h	i	key
w	v	y	f	1	9	9	3	8	proi

(۱)

(الف)



ب) (۱) پیدا کردن max در بین proi های آرایه و قرار دادن آن به عنوان ریشه (Heap Max)

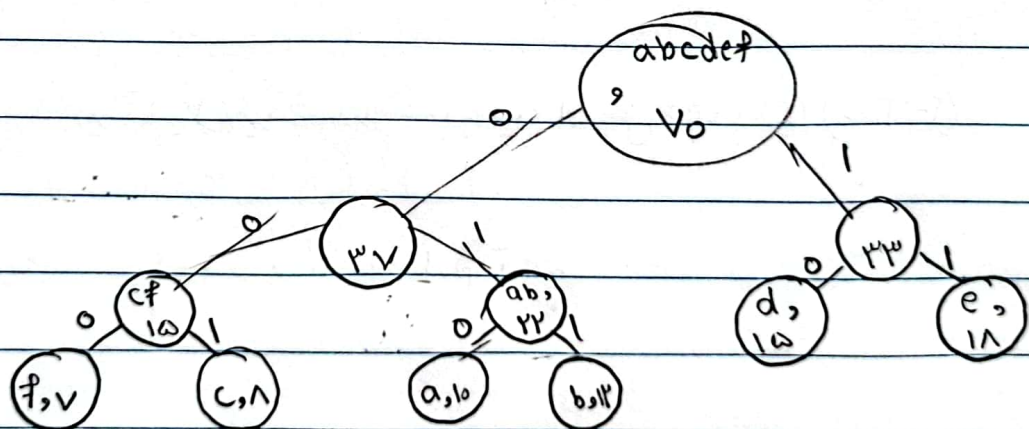
(۲) تقسیم آرایه به ۲ زیر آرایه (زیر درخت راست و چپ) از محل index (BTS)

(۳) تکرار مرحله برای ۲ زیر آرایه تولید شده تا اتمام آرایه

(۲)

~~a: 10~~ ~~b: 11~~ ~~c: 11~~ ~~d: 10~~ ~~e: 11~~ ~~f: 11~~

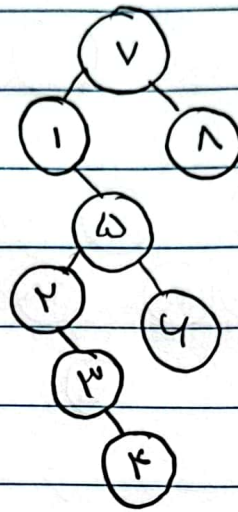
(3)



a: 010 b: 011 c: 001 d: 10 e: 11 f: 000

$$\Rightarrow \text{require bits: } (3 \times 10) + (3 \times 11) + (1 \times 11) + (1 \times 10) + (1 \times 11) + (3 \times 11) \\ = \underline{\underline{116 \text{ bits}}}$$

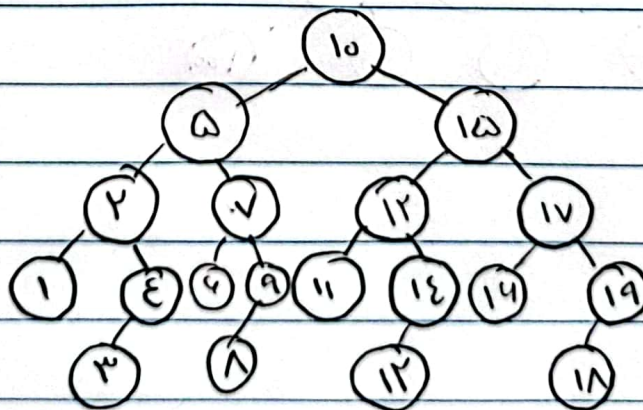
Post order : ۴ ۳ ۲ ۹ ۵ ۱ ۸ ۷ L R Root / BTS (4)



(5) مقدار بیت بیشتری میگیرد چون هر هر سری یک بیت افزایش داریم در صورتی که برای هافمن

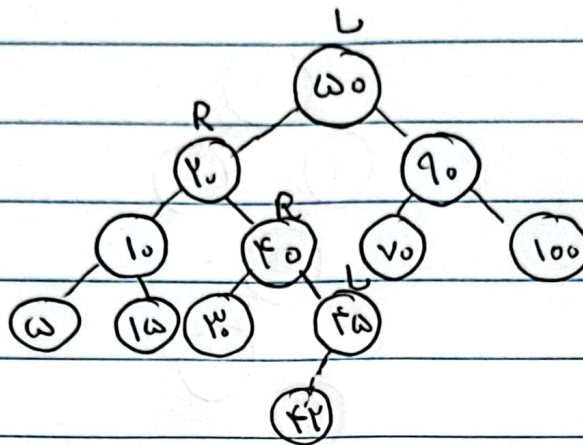
فصل است همه داده یک مقدار بیت داشته باشند

(6) node 19 ← حداقل ارتفاع ۴



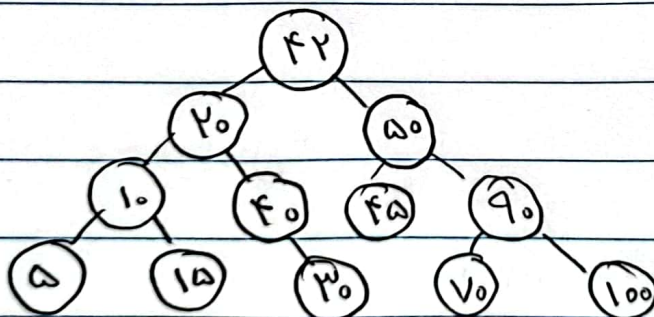
Postorder: 5 15 10 30 45 40 20 70 100 90 50

(7)



الف. بله ب. چون 42 در جای مشخص شده قرار میگیرد ← AVL نیست

ج. ابتدا R (45) و سپس L (40) پس L (20) و سپس R (50)
 LR



(1)

$O(n) \leftarrow \text{inorder traversal} \rightarrow O(k)$ (2)

(3)