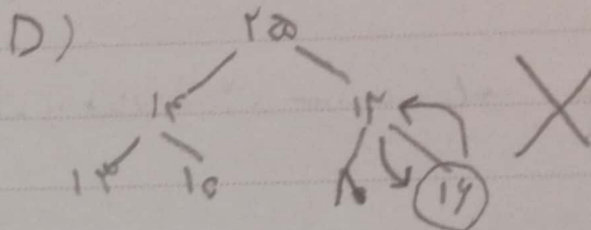
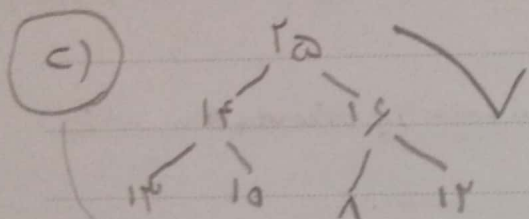
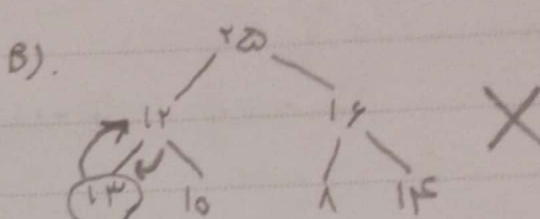
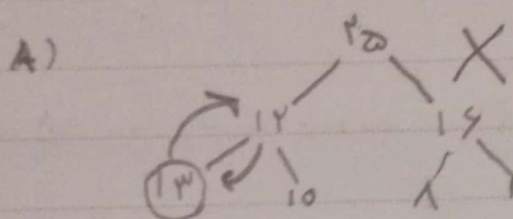
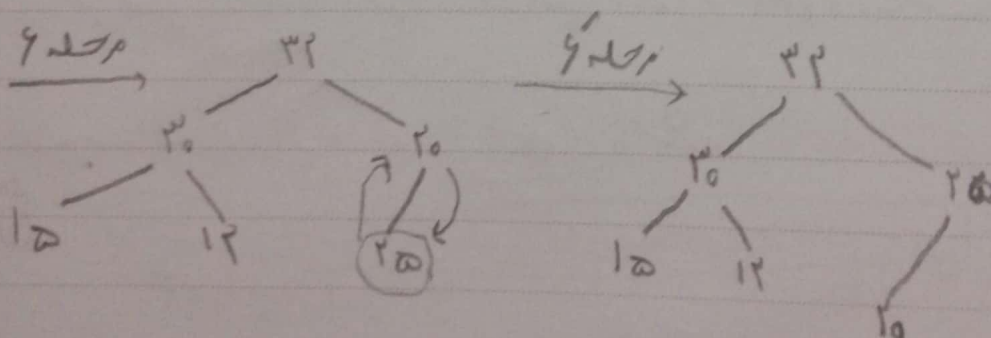
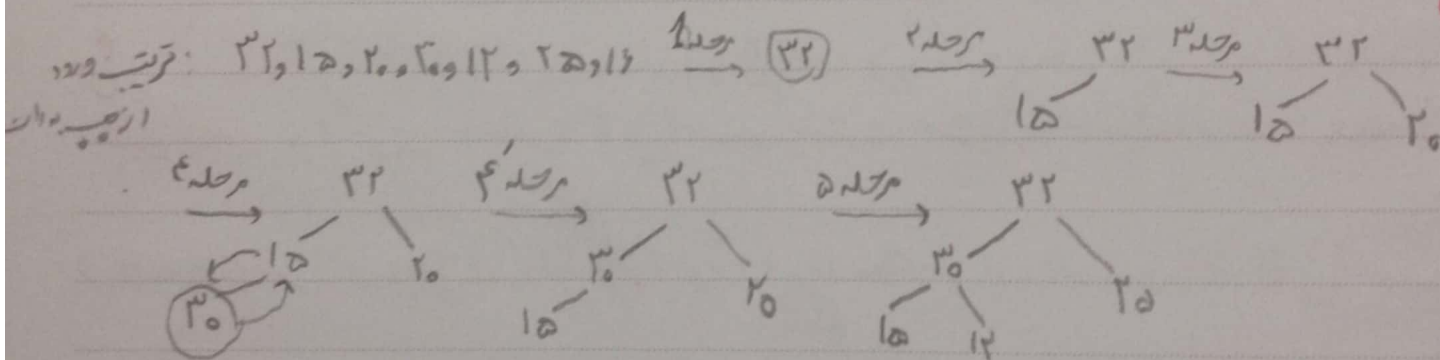


عنوان

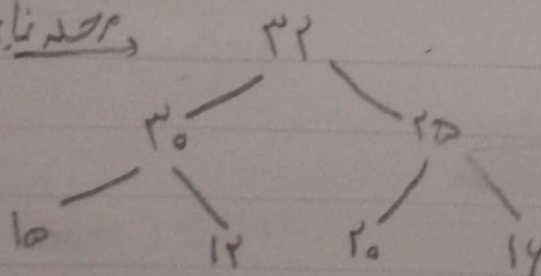


در A و B قانون این که والد از child حمایت می‌کند است رعایت نشده. گزینش درست

سے



محمد ثناء



(a) جواب ہائی

س ۱۰ هفتمین عنصر کوچک باید در رده ۷ام لا سیم اول موجود باشد، با توجه به آنکه binary heap در گره ۲ child دارد، داریم

$$1 + 2 + 4 + 8 + \dots + 2^v = \sum_{i=0}^{v=7} 2^i = 127$$

که عدد ثابت پس درجه آر در مد نظر از $\Theta(1)$ می باشد و گزین $\frac{1}{2}$ درست

ب) اگر min-heap دارای duplicate (تکراری) پیچیدگی آن از آر در $\Theta(\log n)$ باشد

اگر min-heap در آن دسترسی به شاخه های ریشه را نداشته باشیم، پیچیدگی متفاوتی خواهیم داشت

که از آر در $\Theta(\log n)$ می باشد

س ۱۱ برای یافتن کوچکترین عنصر، کافی است سمت چپ شود که همیشه در max heap، والد

(Parent) از شاخه های (childs) بزرگتر می باشد و اگر والد آن خود در نوبه کافی است
(در واقع)

روی آخر عمق (depth) از دو شاخه اصلی معادل به Root، مشابه انجام شده که این عمل

با توجه به مشابه هر کدام با هم از آر در $\Theta(n)$ می باشد

س ۱۲ با فرض اینکه هدف سوال دادن یک آرایه و heap شدن آن و Sort است

func heapSort(int[] array) array.Length = t

for (i = t/2 - 1; i >= 0; i--)

heap(array, t, i)

ساخت Heap

for (i = t - 1; i > 0; i--)

swap array[0] و array[i] together

حرکت به Root

heap(array, i, 0)

func Heap (int[] array, int m, int i)

Left = $2i + 1$

~~q~~ = 1

Right = $2i + 2$

Root & child & index of low

if (Left < m and array[Left] > array[i])

q = Left

{check it for right} if (Right < m and array[Right] > array[q])

also

q = Right

if (q not largest num \rightarrow not root?)

swap array[i] and array[q]

Heap(array, m, q)

در Heap چک می‌کنیم که عدد ورودی در جایگاه درست باشد یا نه. اگر بزرگتر باشد