

DATE: / /

SUBJECT:

BEGIN THE BEST QUALITY PAPER

Quiz (1):

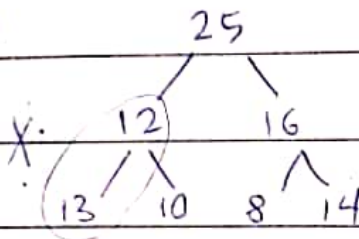
node سے children max-heap tree

1 Array سے tree children

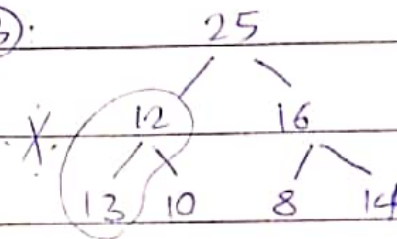
(2nd) index (i) index node heap tree

index (2i+2) index

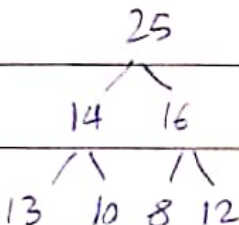
(A)



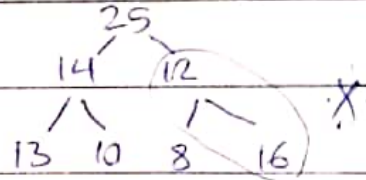
(B)



(C)



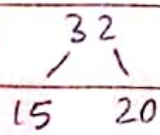
(D)



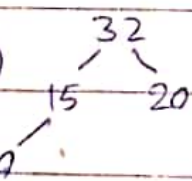
پس ترتیباً درخت ترسیم می باشد

Quiz 2 : 32, 15, 20, 30, 12, 25, 16

(1)



(2)



ابتدا node اصلی ترسیم می کنیم

و پس 30 به tree اضافه می کنیم

و پس 15 از 30 بزرگتر است پس چپ می بینیم

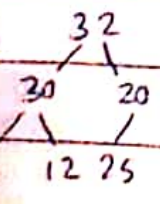
و چون 12 کوچکتر است پس 12 به tree insert می کنیم

و در سطح 30 هم 25 چپ می بینیم پس 25 به tree اضافه می کنیم

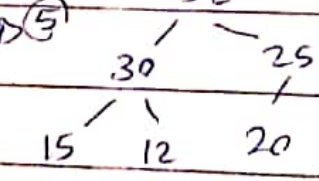
و پس 16 را هم 20 چپ می بینیم پس 16 به tree اضافه می کنیم

و پس 16 را هم 25 چپ می بینیم

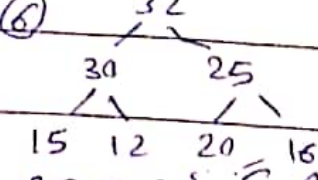
(4)



(5)



(6)



BEGIN

پس ترتیباً درخت ترسیم می باشد

Quiz ④ leaves, main heap, worst case, node, complexity $O(n)$

Quiz ③: duplicate, first, second, 3rd, 4th smallest, $\frac{6}{2} = 3$, $i=0$

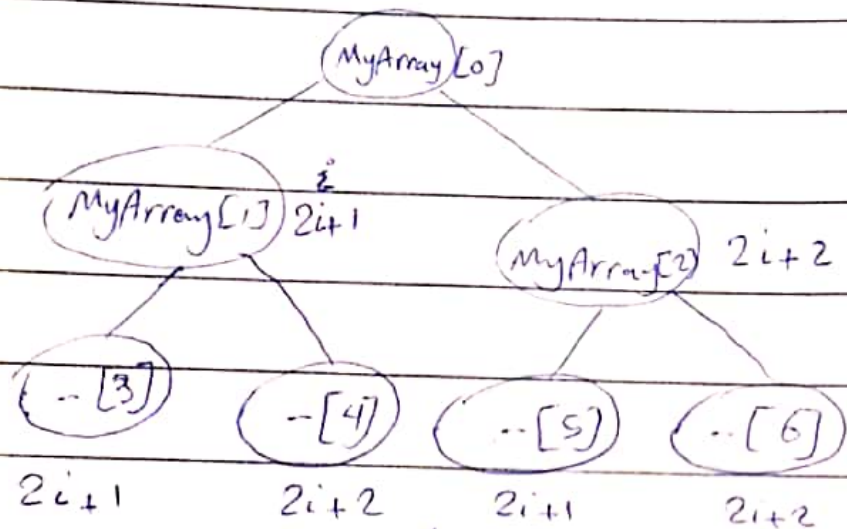
operation, $O(1)$, $O(1)$ times, $O(1)$

duplicate, node, $O(n \log n)$

$1 + 2 + 4 + 8 + 16 + 32 + 64 = O(\log 7) = O(1)$ times
 $1 + 2 + 4 + 8 + \dots + 2^{n-1} = O(n \log n)$

Quiz (5)

ابتدا $2^n - 1$ را از یک بزرگترین 2^n که در $MyArray$ میگردد



internal node

tree

for (int k=0; k < $\log_2 n$; k++)

{ if (MyArray[k] < MyArray[2k+1])

{ int s = MyArray[k];

MyArray[k] = MyArray[2k+1];

MyArray[2k+1] = s;

else if (MyArray[k] < MyArray[2k+2])

{ int s = MyArray[k] > MyArray[2k+2];

MyArray[2k+2] = MyArray[k];

MyArray[k] = s;

}

این عملیات را برای هر یک از گره های داخلی (internal node) انجام دهیم

این عملیات را برای هر یک از گره های داخلی (internal node) انجام دهیم

internal node: $2^0 + 2^1 + \dots + 2^{n-2} = 2^{n-1} - 1$