

## پاسخ تمرین شماره ۳



ساختمان داده - پاییز ۱۳۹۸

دانشکده مهندسی برق و کامپیوتر

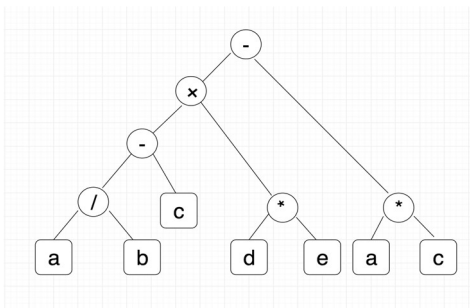
مسئول تمرین : غزل مینایی

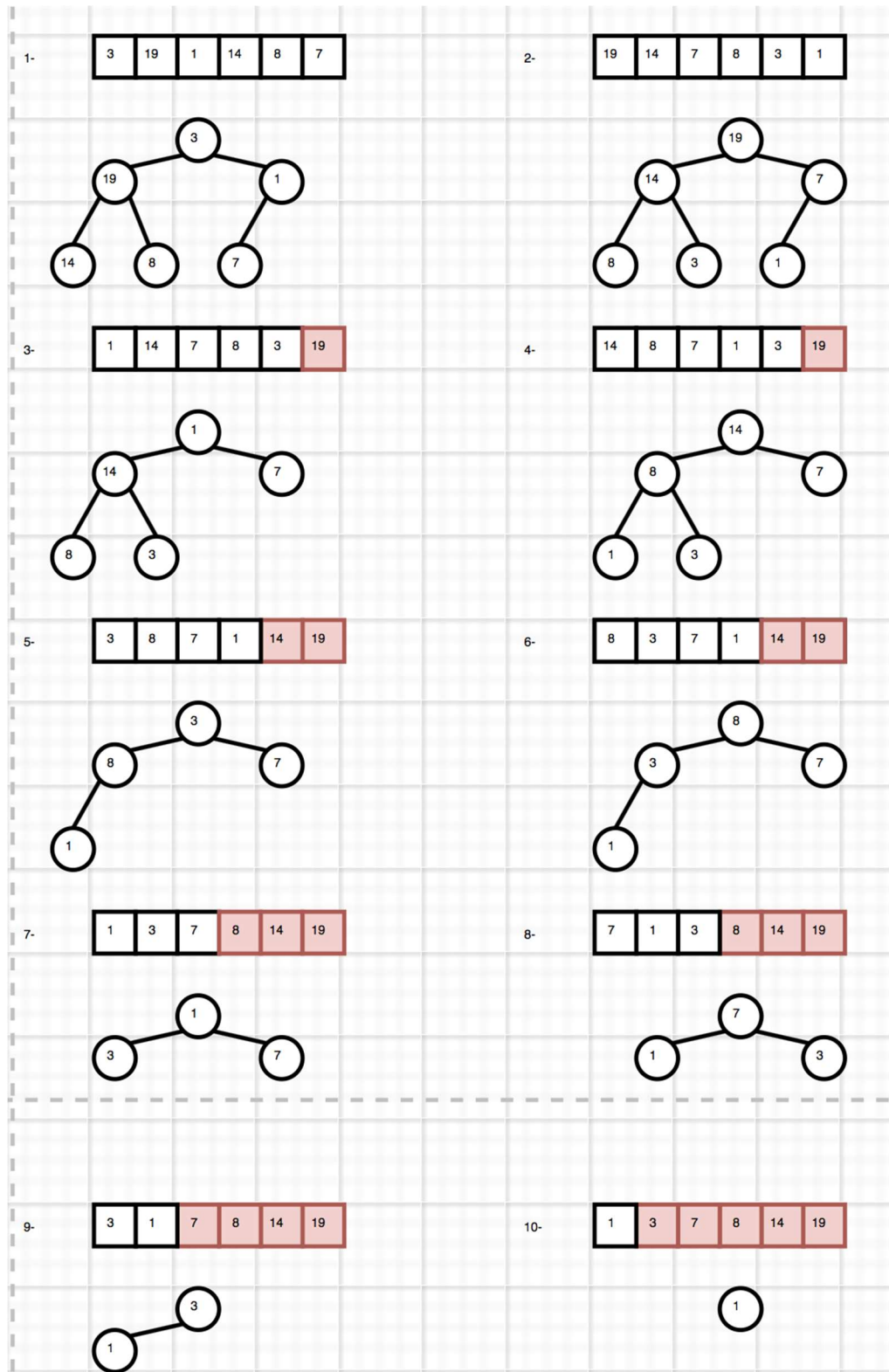
استاد : هشام فیلی

۱- الگوریتم چاپ میان ترتیب از روی درخت عبارت.

```
inOrder(node tree):  
  if (tree != NULL)  
    inOrder(tree->right)  
    print(tree->data)  
    inOrder(tree->left)
```

$a/b-c+d*e-a*c$





۳- مرحله اول :

میدانیم در heap با تعداد n عنصر حداکثر  $\left\lfloor \frac{n}{2^{h+1}} \right\rfloor$  عنصر با ارتفاع h وجود دارد . پس هزینه ی کلی ساخت درخت heap برابر است با:

$$\begin{aligned} T(n) &= \sum_{h=0}^{\lg(n)} \left\lfloor \frac{n}{2^{h+1}} \right\rfloor * O(h) \\ &= O(n * \sum_{h=0}^{\lg(n)} \frac{h}{2^h}) \\ &= O(n * \sum_{h=0}^{\infty} \frac{h}{2^h}) \end{aligned}$$

مرحله دوم :

می دانیم :

$$\begin{aligned} \sum_{n=0}^{\infty} x^n &= \frac{1}{1-x} \\ \sum_{n=0}^{\infty} nx^n &= \frac{x}{(1-x)^2} \end{aligned}$$

پس :

$$\begin{aligned} &= O(n * \frac{\frac{1}{2}}{(1-\frac{1}{2})^2}) \\ &= O(n * 2) \\ &= O(n) \end{aligned}$$

۴- از یک min heap استفاده می کنیم. در هر خانه از هیپ یک جفت شامل داده و اولویت نگه می داریم اعداد را به ترتیب ورود از یک تا n اولویت می دهیم. همیشه اولین عدد خارج شده از سر هیپ اولین عدد وارد شده خواهد بود.

heap = Heap()

n = 0

enqueue(x):

    p = pair(x, n)

    n = n+1

    heap.enqueueHeap(p)

dequeue():

    p = heap.dequeue()

    n = n-1

    return p(2)

- 5

level(Node root, Node x, int index, int level){

    if (root == NULL or level

        return

    if (root == x)

        level = index

    level(root->left, x, index + 1, level)

    level(root->right, x, index + 1, level)

}

printLevel(Node root, Node node, int level){

    if (root == NULL)

        return

    if (level == 1){

        print (root->data)

        return

    }

    if (!(root->left and root->left == node or root->right and root->right == node)){

        printLevel(root->left, node, level - 1);

        printLevel(root->right, node, level - 1);

```

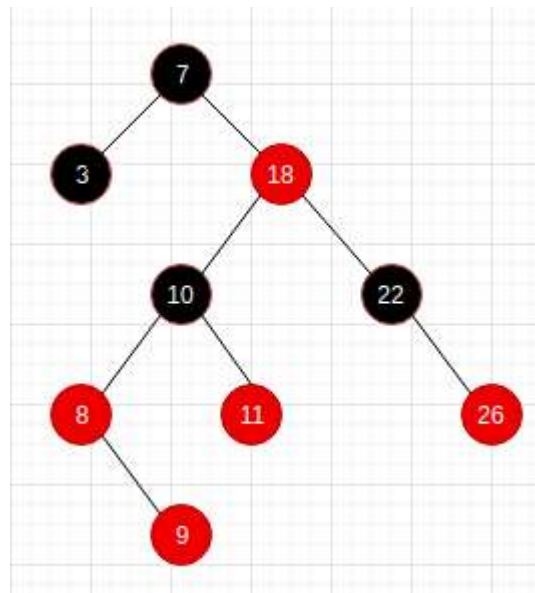
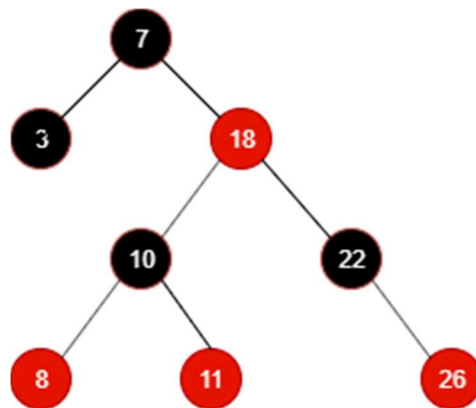
    }
}

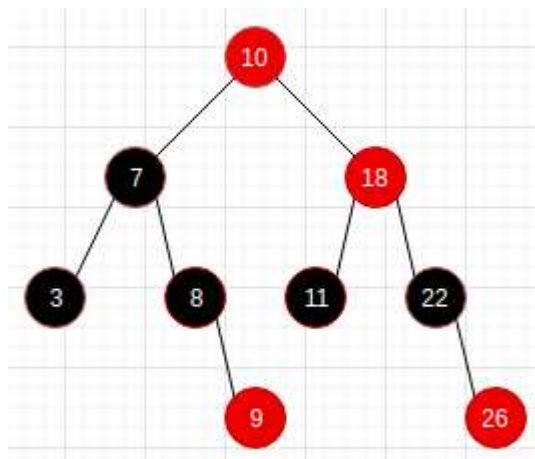
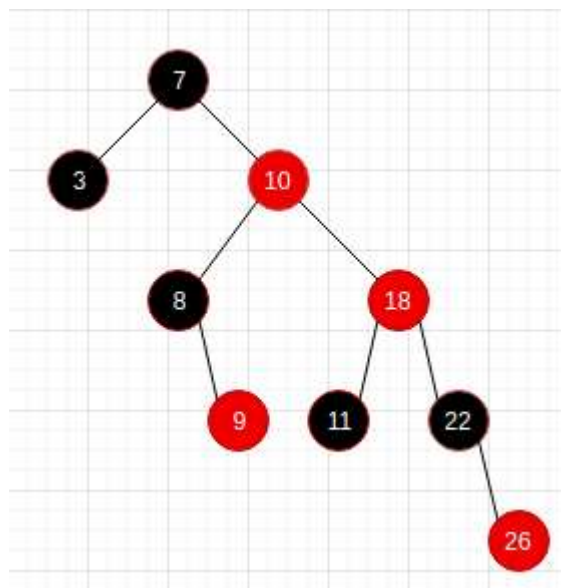
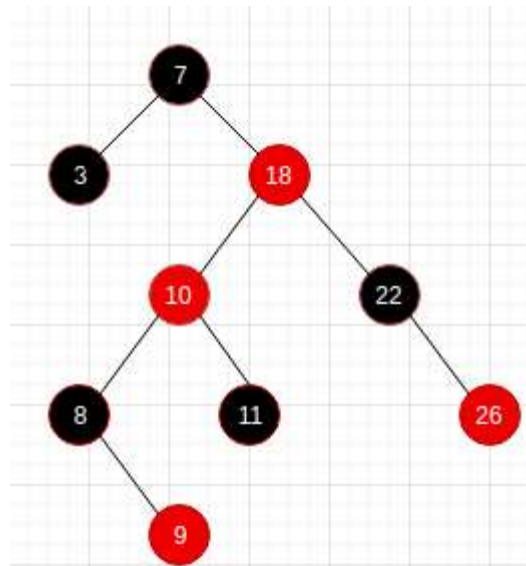
printAllCousins(Node root, Node node){
    int level = 0;
    level(root, node, 1, level);
    printLevel(root, node, level);
}

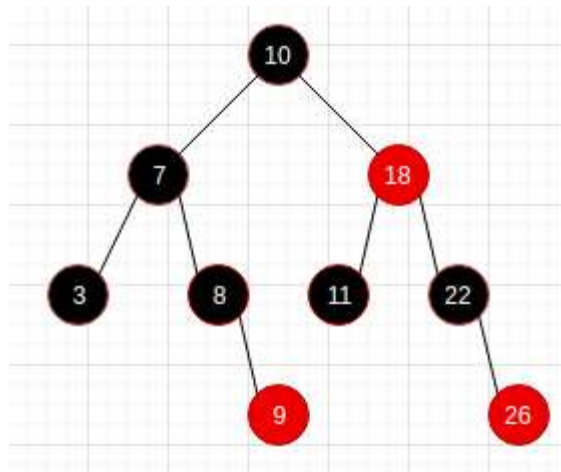
```

-۶

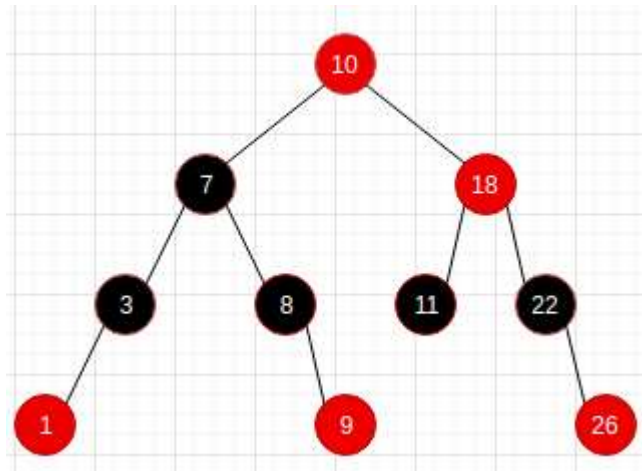
A.



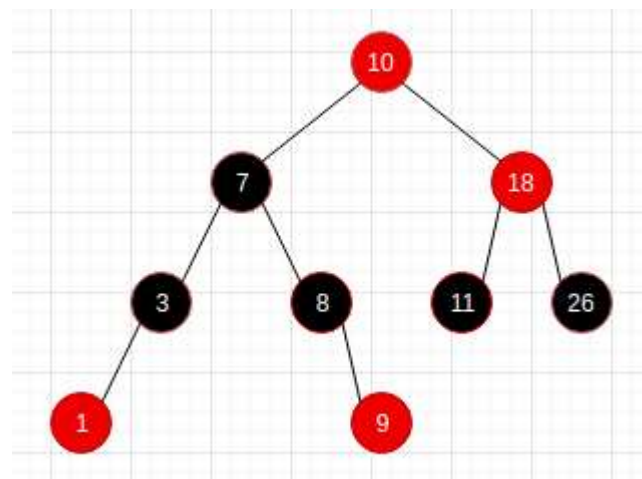
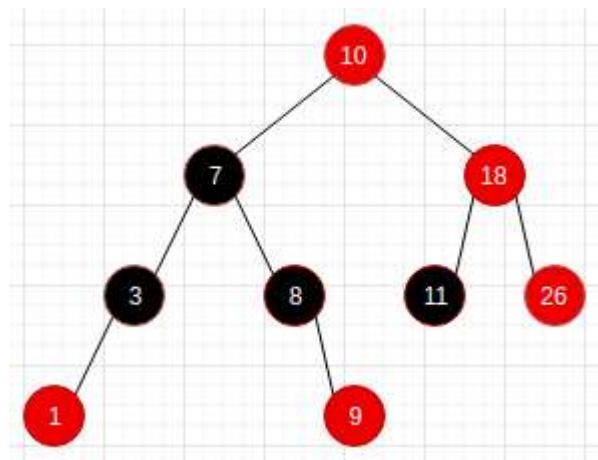




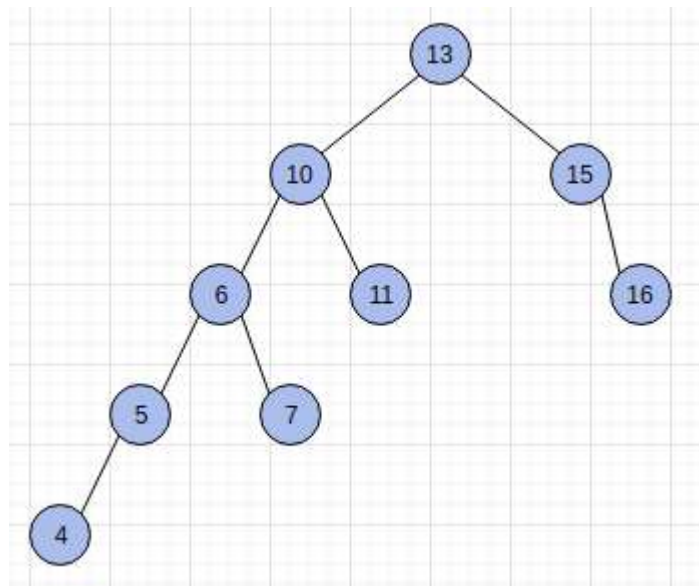
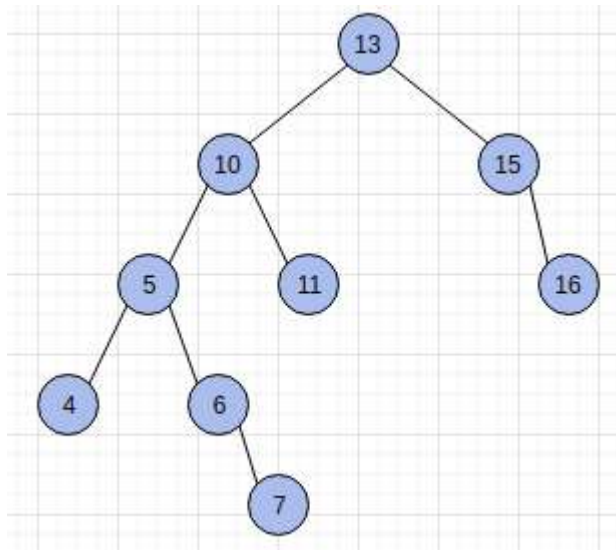
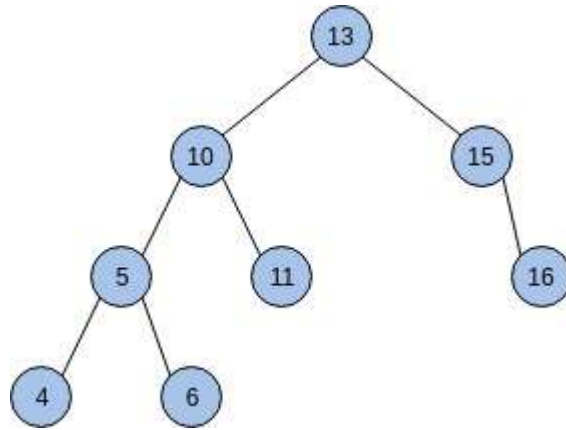
B.

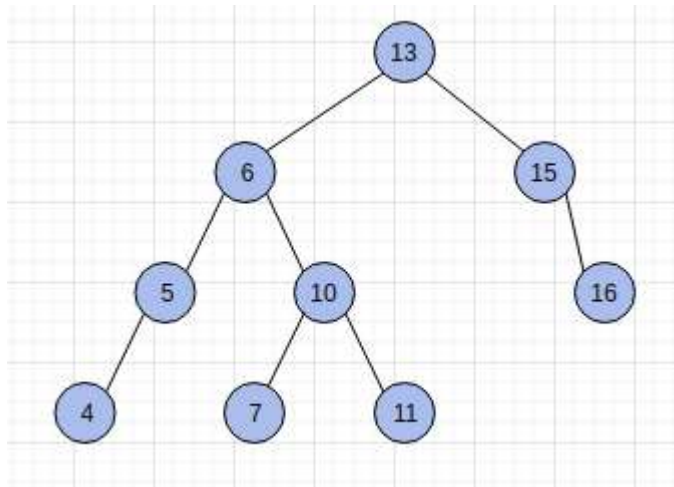


C.









۸- کوتاه ترین فاصله  $x$  تا برگ :  $d_x$  و ارتفاع را هم  $h_x$  در نظر میگیریم. داریم که  $2/Bh_x > h_x$  در غیر این صورت حداقل دو راس قرمز مجاور میشوند. میدانیم که  $d_x > bh_x$  پس خواهیم داشت :  $d \geq h/2$