

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.left = None

self.right = None

self.parent = None

def inorderTraversal(root):

current = root

stack = []

while True:

if current is not None:

stack.append(current)

current = current.left

elif stack:

current = stack.pop()

print(current.data)

current = current.right

else:

break

a = 111

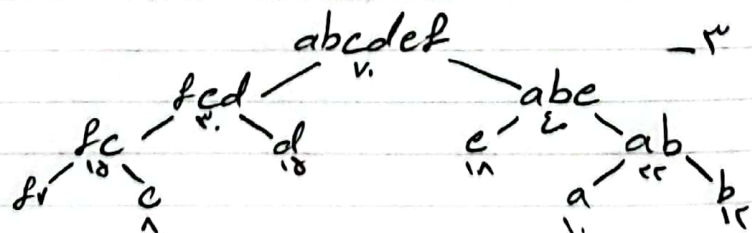
d = 1

b = 111

e = 1

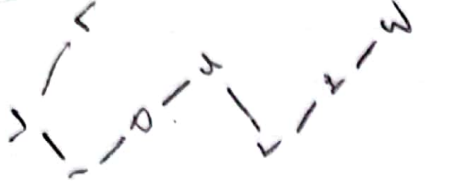
c = 100

f = 1000

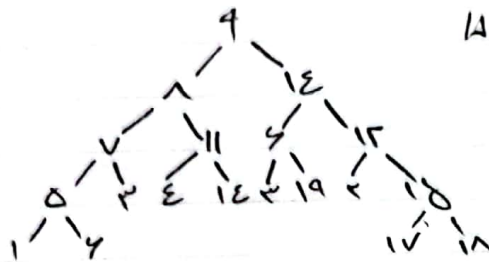


410.72.77

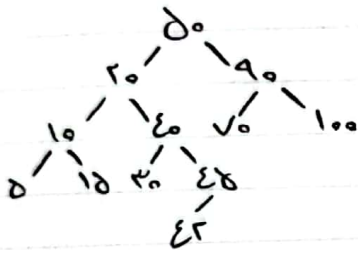
ملک دور کرد



۵- تعداد کل بیت های لازم در این روش بیست و نوار دسی هافمن است.



۱۲ ع



(۳) ضد

(۳) دوران بہت اسے (۲، ۱)

۱-  $2^{n-2}$  درخت

$\epsilon O(r^n)$  -4

-10

class made:

```
def __init__(self, data):
```

self.root = root

self: right - right

self. left. left

def cheek (root):

if root.left > root or root.right < root:

return False

return True

$90(n^2) \quad \underline{\underline{-11}}$