

## به نام خدا

ابجکت plt از کتابخانه matplotlib برای ایجاد ظاهر درخت استفاده می کنیم.  
ابجکت nx از کتابخانه networkx برای ایجاد مختصات گره های درخت استفاده می کنیم.

```
import matplotlib.pyplot as plt
import networkx as nx
```

کلاس Node برای ایجاد درخت باینری

```
class Node:
    def __init__(self, data):
        self.left = None
        self.right = None
        self.value = data
```

تابع insert برای اضافه کردن ارایه به درخت باینری (این درخت به صورت درخت باینری کامل پرمی شود).

```
def insert(arr, root, i, length):
    if i < length:
        if arr[i] is not None:
            temp = Node(arr[i])
            root = temp
            root.left = insert(arr, root.left, 2 * i + 1, length)
            root.right = insert(arr, root.right, 2 * i + 2, length)
    return root
```

توابع preorder و inorder و postorder به ترتیب برای نمایش پیمایش های پیش سری و درون سری و پس سری استفاده می شود.

```
def preorder(root):  
    if root:  
        print(root.value, end=' ')  
        preorder(root.left)  
        preorder(root.right)
```

```
def inorder(root):  
    if root:  
        inorder(root.left)  
        print(root.value, end=' ')  
        inorder(root.right)
```

```
def postorder(root):  
    if root:  
        postorder(root.left)  
        postorder(root.right)  
        print(root.value, end=' ')
```

تابع `add_edges` برای ایجاد تصویر گرافیکی گره های های درخت که از کتابخانه های `matplotlib` و `networkx` استفاده می کند.

برای اضافه کردن فرزندان چپ و راست خود تابع درون خودش `call` می شود .

```
def add_edges(G, root, pos={}, x=0, y=0, layer=1):
    if root:
        pos[root.value] = (x, y)
        if root.left:
            G.add_edge(root.value, root.left.value)
            l = x - 1 / layer
            add_edges(G, root.left, pos, x=l, y=y - 1, layer=layer + 1)
        if root.right:
            G.add_edge(root.value, root.right.value)
            r = x + 1 / layer
            add_edges(G, root.right, pos, x=r, y=y - 1, layer=layer + 1)
    return G, pos
```

شروع برنامه از اینجااست:

ایجاد آرایه برای ذخیره گره های درخت

دریافت ورودی از کاربر:

۱. اگر ورودی کلمه `exit` باشد دریافت ورودی را از کاربر متوقف می کند.

۲. اگر داده ورودی وجود داشته باشد به آرایه اضافه می شود

۳. اگر داده وجود نداشته باشد ( یعنی `space` ورودی باشد ) در آرایه `none` ذخیره می شود.

```
data = []
while True:
    d = input("Enter data for tree (for finish enter 'exit'): ").strip()
    if d == 'exit':
        break
    if d:
        data.append(d)
    else:
        data.append(None)
```

ارسال ارایه data برای تابع insert برای ایجاد درخت باینری

```
root = None  
root = insert(data, root, 0, len(data))
```

ایجاد محیطی برای تشکیل درخت به صورت گرافیکی

```
G = nx.Graph()
```

ایجاد یال بین گره های درخت با تابع add\_edges

```
G, pos = add_edges(G, root)
```

اتصال گره ها و یال های تشکیل شده در تابع add\_edges و ایجاد تصویر گرافیکی

```
fig, ax = plt.subplots()  
nx.draw(G, pos, with_labels=True, node_size=500, node_color="skyblue", ax=ax)
```

نمایش خروجی برنامه :

۱. ایجاد حلقه برای نمایش نامحدود خروجی برنامه (inorder , preorder , postorder و نمایش تصویر گرافیکی درخت)

۲. اگر ورودی preorder باشد : پیمایش پیش سری درخت را نشان می دهد.

۳. اگر ورودی inorder باشد : پیمایش درون سری درخت را نشان می دهد.

۴. اگر ورودی postorder باشد : پیمایش پس سری درخت را نشان می دهد.

۵. اگر ورودی show باشد : تصویر گرافیکی درخت را نمایش می دهد

۶. اگر ورودی exit باشد : از برنامه خارج می شود.

۷. اگر کلمه ورودی غیر از این کلمات باشد از کاربر دوباره سوال می شود

تا زمانی کلمه exit وارد نشود از کاربر دوباره نمایش خروجی سوال می شود

```
while True:
    ans = input("Enter type of navigation (preorder, inorder, postorder and for finish enter 'exit'): ")

    if ans in ['preorder', 'inorder', 'postorder', 'show', 'exit']:
        if ans == 'preorder':
            print("PreOrder Navigation: ", end='')
            preorder(root)
            print()
        elif ans == 'inorder':
            print("InOrder Navigation: ", end='')
            inorder(root)
            print()
        elif ans == 'postorder':
            print("PostOrder Navigation: ", end='')
            postorder(root)
            print()
        elif ans == 'show':
            plt.show()
        elif ans == 'exit':
            break
    else:
        print("Please enter a valid order!")
```

اعضای گروه :

سید محمد علی موسوی قلی آباد

سید محمد مهدیار حامد نعمت پور

ریحانه اسدی