In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sea
```

In [2]:
```
df = pd.read_csv(r"C:\Users\user\Downloads\C10_loan1.csv")
df
```

Out[2]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	Yes	Single	125	No
1	No	Married	100	No
2	No	Single	70	No
3	Yes	Married	120	No
4	No	Divorced	95	Yes
5	No	Married	60	No
6	Yes	Divorced	220	No
7	No	Single	85	Yes
8	No	Married	75	No
9	No	Single	90	Yes

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Home Owner          10 non-null     object
 1   Marital Status      10 non-null     object
 2   Annual Income       10 non-null     int64 
 3   Defaulted Borrower  10 non-null     object
dtypes: int64(1), object(3)
memory usage: 448.0+ bytes
```

In [4]:
```
df2 = df[['Home Owner','Marital Status','Annual Income']]
```

In [5]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Home Owner      10 non-null     object
 1   Marital Status  10 non-null     object
 2   Annual Income   10 non-null     int64 
dtypes: int64(1), object(2)
memory usage: 368.0+ bytes
```

In [7]:
```
ch = {"Home Owner":{'Yes':1,'No':0},"Marital Status":{'Single':1,'Married':2,'Divorced':3}}
df2 = df2.replace(ch)
df2
```

Out[7]:

	Home Owner	Marital Status	Annual Income
0	1	1	125
1	0	2	100
2	0	1	70
3	1	2	120
4	0	3	95
5	0	2	60
6	1	3	220
7	0	1	85
8	0	2	75
9	0	1	90

In [8]:
```
df2['Home Owner'].value_counts()
```

Out[8]:
```
0    7
1    3
Name: Home Owner, dtype: int64
```

In [9]:
```
df2.columns
```

Out[9]:
```
Index(['Home Owner', 'Marital Status', 'Annual Income'], dtype='object')
```

In [10]:
```
x = df2[['Marital Status', 'Annual Income']]
y = df2['Home Owner']
```

In [11]:
```
from sklearn.model_selection import train_test_split
```

In [12]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.70)
```

In [13]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [14]:
```
rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[14]:
```
RandomForestClassifier()
```

In [15]:
```
parameters = {
    'max_depth':[1,2,3,4,5],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [16]:
```
from sklearn.model_selection import GridSearchCV
```

In [17]:
```
grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='accuracy')
grid_search.fit(x_train,y_train)
```

Out[17]:
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 10, 15, 20, 25],
                         'n_estimators': [10, 20, 30, 40, 50]},
             scoring='accuracy')
```

In [18]:
```
grid_search.best_score_
```

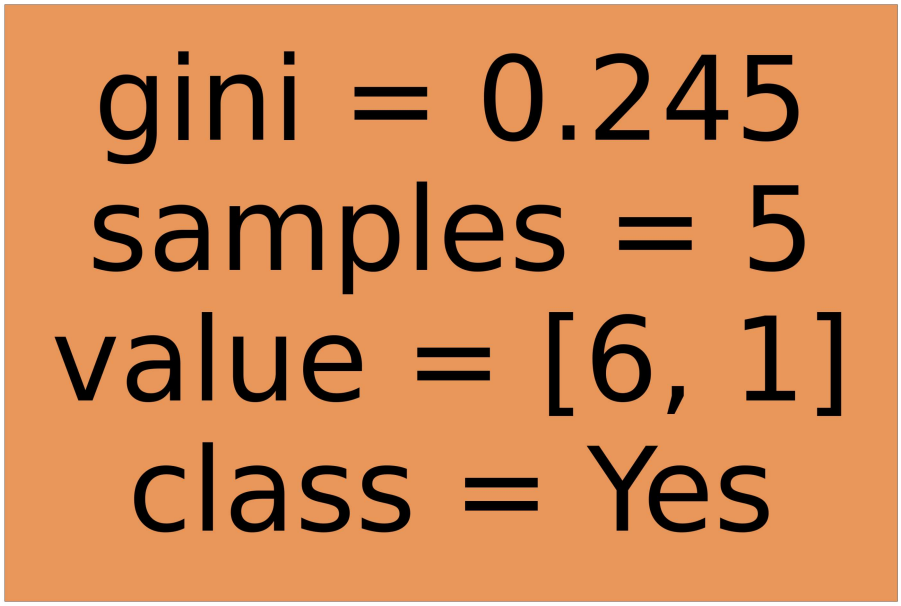
Out[18]:
```
0.7083333333333333
```

In [19]:
```
from sklearn.tree import plot_tree
```

In [20]:
```
rfc_best= grid_search.best_estimator_
```

In [21]:
```
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=["Yes","No"],filled=True
```

Out[21]:
```
[Text(2232.0, 1087.2, 'gini = 0.245\nsamples = 5\nvalue = [6, 1]\nclass = Yes')]
```



In [ ]:
```

```