In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier

```

In [2]:
```
traindf=pd.read_csv(r"C:\Users\P. VIJAY KUMAR\Downloads\Mobile_Price_Classification_train.csv")
traindf
```

Out[2]:

| | battery_power | blue | clock_speed | dual_sim | fc | four_g | int_memory | m_dep | mobile_wt | n_cores | ... | px_height | px_width | ram | sc_h |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 842 | 0 | 2.2 | 0 | 1 | 0 | 7 | 0.6 | 188 | 2 | ... | 20 | 756 | 2549 | 9 |
| **1** | 1021 | 1 | 0.5 | 1 | 0 | 1 | 53 | 0.7 | 136 | 3 | ... | 905 | 1988 | 2631 | 17 |
| **2** | 563 | 1 | 0.5 | 1 | 2 | 1 | 41 | 0.9 | 145 | 5 | ... | 1263 | 1716 | 2603 | 11 |
| **3** | 615 | 1 | 2.5 | 0 | 0 | 0 | 10 | 0.8 | 131 | 6 | ... | 1216 | 1786 | 2769 | 16 |
| **4** | 1821 | 1 | 1.2 | 0 | 13 | 1 | 44 | 0.6 | 141 | 2 | ... | 1208 | 1212 | 1411 | 8 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1995** | 794 | 1 | 0.5 | 1 | 0 | 1 | 2 | 0.8 | 106 | 6 | ... | 1222 | 1890 | 668 | 13 |
| **1996** | 1965 | 1 | 2.6 | 1 | 0 | 0 | 39 | 0.2 | 187 | 4 | ... | 915 | 1965 | 2032 | 11 |
| **1997** | 1911 | 0 | 0.9 | 1 | 1 | 1 | 36 | 0.7 | 108 | 8 | ... | 868 | 1632 | 3057 | 9 |
| **1998** | 1512 | 0 | 0.9 | 0 | 4 | 1 | 46 | 0.1 | 145 | 5 | ... | 336 | 670 | 869 | 18 |
| **1999** | 510 | 1 | 2.0 | 1 | 5 | 1 | 45 | 0.9 | 168 | 6 | ... | 483 | 754 | 3919 | 19 |

2000 rows × 21 columns

In [3]:
```
testdf=pd.read_csv(r"C:\Users\P. VIJAY KUMAR\Downloads\Mobile_Price_Classification_test.csv")
testdf
```

Out[3]:

| | id | battery_power | blue | clock_speed | dual_sim | fc | four_g | int_memory | m_dep | mobile_wt | ... | pc | px_height | px_width | ram | sc_h |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1043 | 1 | 1.8 | 1 | 14 | 0 | 5 | 0.1 | 193 | ... | 16 | 226 | 1412 | 3476 | 12 |
| **1** | 2 | 841 | 1 | 0.5 | 1 | 4 | 1 | 61 | 0.8 | 191 | ... | 12 | 746 | 857 | 3895 | 6 |
| **2** | 3 | 1807 | 1 | 2.8 | 0 | 1 | 0 | 27 | 0.9 | 186 | ... | 4 | 1270 | 1366 | 2396 | 17 |
| **3** | 4 | 1546 | 0 | 0.5 | 1 | 18 | 1 | 25 | 0.5 | 96 | ... | 20 | 295 | 1752 | 3893 | 10 |
| **4** | 5 | 1434 | 0 | 1.4 | 0 | 11 | 1 | 49 | 0.5 | 108 | ... | 18 | 749 | 810 | 1773 | 15 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **995** | 996 | 1700 | 1 | 1.9 | 0 | 0 | 1 | 54 | 0.5 | 170 | ... | 17 | 644 | 913 | 2121 | 14 |
| **996** | 997 | 609 | 0 | 1.8 | 1 | 0 | 0 | 13 | 0.9 | 186 | ... | 2 | 1152 | 1632 | 1933 | 8 |
| **997** | 998 | 1185 | 0 | 1.4 | 0 | 1 | 1 | 8 | 0.5 | 80 | ... | 12 | 477 | 825 | 1223 | 5 |
| **998** | 999 | 1533 | 1 | 0.5 | 1 | 0 | 0 | 50 | 0.4 | 171 | ... | 12 | 38 | 832 | 2509 | 15 |
| **999** | 1000 | 1270 | 1 | 0.5 | 0 | 4 | 1 | 35 | 0.1 | 140 | ... | 19 | 457 | 608 | 2828 | 9 |

1000 rows × 21 columns

In [4]:
```
1 traindf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   battery_power  2000 non-null   int64  
 1   blue           2000 non-null   int64  
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64  
 4   fc             2000 non-null   int64  
 5   four_g         2000 non-null   int64  
 6   int_memory     2000 non-null   int64  
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64  
 9   n_cores        2000 non-null   int64  
 10  pc             2000 non-null   int64  
 11  px_height      2000 non-null   int64  
 12  px_width       2000 non-null   int64  
 13  ram            2000 non-null   int64  
 14  sc_h           2000 non-null   int64  
 15  sc_w           2000 non-null   int64  
 16  talk_time      2000 non-null   int64  
 17  three_g        2000 non-null   int64  
 18  touch_screen   2000 non-null   int64  
 19  wifi           2000 non-null   int64  
 20  price_range    2000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

In [5]:
```
1 testdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             1000 non-null   int64  
 1   battery_power  1000 non-null   int64  
 2   blue           1000 non-null   int64  
 3   clock_speed    1000 non-null   float64
 4   dual_sim       1000 non-null   int64  
 5   fc             1000 non-null   int64  
 6   four_g         1000 non-null   int64  
 7   int_memory     1000 non-null   int64  
 8   m_dep          1000 non-null   float64
 9   mobile_wt      1000 non-null   int64  
 10  n_cores        1000 non-null   int64  
 11  pc             1000 non-null   int64  
 12  px_height      1000 non-null   int64  
 13  px_width       1000 non-null   int64  
 14  ram            1000 non-null   int64  
 15  sc_h           1000 non-null   int64  
 16  sc_w           1000 non-null   int64  
 17  talk_time      1000 non-null   int64  
 18  three_g        1000 non-null   int64  
 19  touch_screen   1000 non-null   int64  
 20  wifi           1000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 164.2 KB
```

In [6]:
```
1 traindf.shape,testdf.shape
2
```

Out[6]: ((2000, 21), (1000, 21))

In [7]:
```
traindf=traindf.head(1000)
traindf
```

Out[7]:

| | battery_power | blue | clock_speed | dual_sim | fc | four_g | int_memory | m_dep | mobile_wt | n_cores | ... | px_height | px_width | ram | sc_h |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 842 | 0 | 2.2 | 0 | 1 | 0 | 7 | 0.6 | 188 | 2 | ... | 20 | 756 | 2549 | 9 |
| **1** | 1021 | 1 | 0.5 | 1 | 0 | 1 | 53 | 0.7 | 136 | 3 | ... | 905 | 1988 | 2631 | 17 |
| **2** | 563 | 1 | 0.5 | 1 | 2 | 1 | 41 | 0.9 | 145 | 5 | ... | 1263 | 1716 | 2603 | 11 |
| **3** | 615 | 1 | 2.5 | 0 | 0 | 0 | 10 | 0.8 | 131 | 6 | ... | 1216 | 1786 | 2769 | 16 |
| **4** | 1821 | 1 | 1.2 | 0 | 13 | 1 | 44 | 0.6 | 141 | 2 | ... | 1208 | 1212 | 1411 | 8 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **995** | 1456 | 0 | 1.6 | 1 | 5 | 0 | 49 | 0.2 | 193 | 3 | ... | 1285 | 1427 | 3624 | 12 |
| **996** | 774 | 0 | 0.5 | 1 | 2 | 1 | 10 | 0.5 | 188 | 2 | ... | 1480 | 1731 | 2944 | 8 |
| **997** | 1068 | 0 | 0.5 | 1 | 0 | 1 | 19 | 0.9 | 197 | 8 | ... | 322 | 875 | 1209 | 19 |
| **998** | 1373 | 1 | 1.9 | 1 | 1 | 1 | 29 | 0.9 | 141 | 6 | ... | 1220 | 1348 | 2752 | 15 |
| **999** | 1777 | 1 | 3.0 | 0 | 3 | 0 | 20 | 0.6 | 188 | 6 | ... | 511 | 616 | 3868 | 5 |

1000 rows × 21 columns

In [8]:
```
X=testdf
y=traindf['price_range']
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.7,random_state=42)
```

In [9]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
```

Out[9]:
```
RandomForestClassifier()
```

In [10]:
```
rf=RandomForestClassifier()
```

In [11]:
```
params={'max_depth':[2,3,5,10,20],'min_samples_leaf':[5,10,20,50,100,200],'n_estimators':[10,25,30,50,100,200]}
```

In [12]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rf,param_grid=params,cv=2,scoring="accuracy")

```

In [13]:
```
grid_search.fit(X_train,y_train)
```

Out[13]:
```
GridSearchCV
estimator: RandomForestClassifier
RandomForestClassifier
```

In [14]:
```
grid_search.best_score_

```

Out[14]:
```
0.3014285714285714
```

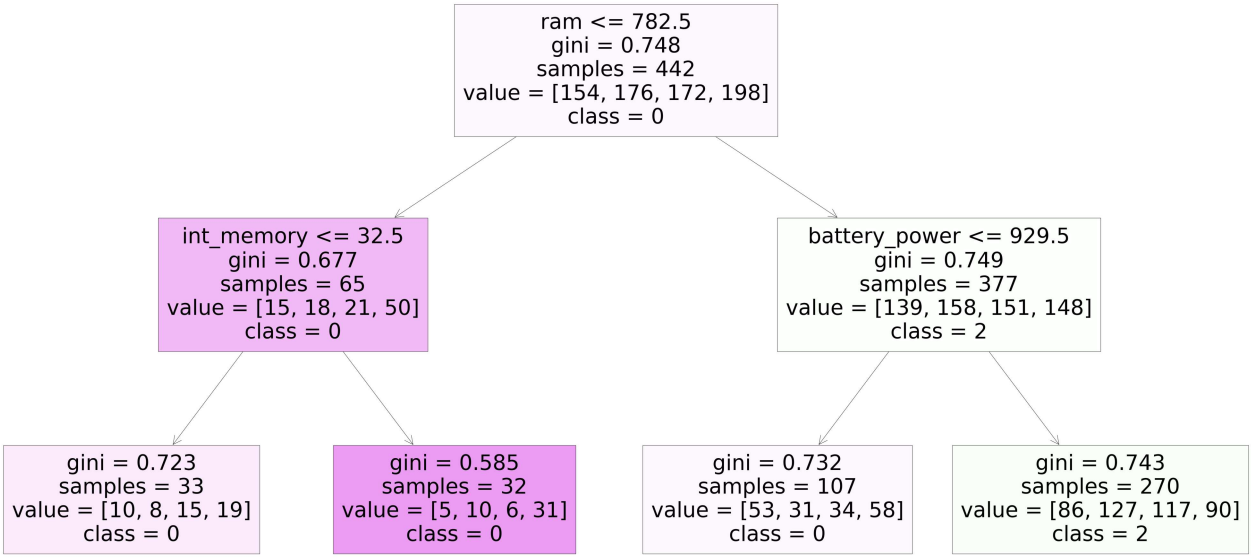
In [15]:
```
rf_best=grid_search.best_estimator_
rf_best
```

Out[15]:
```
RandomForestClassifier(max_depth=2, min_samples_leaf=10, n_estimators=10)
```

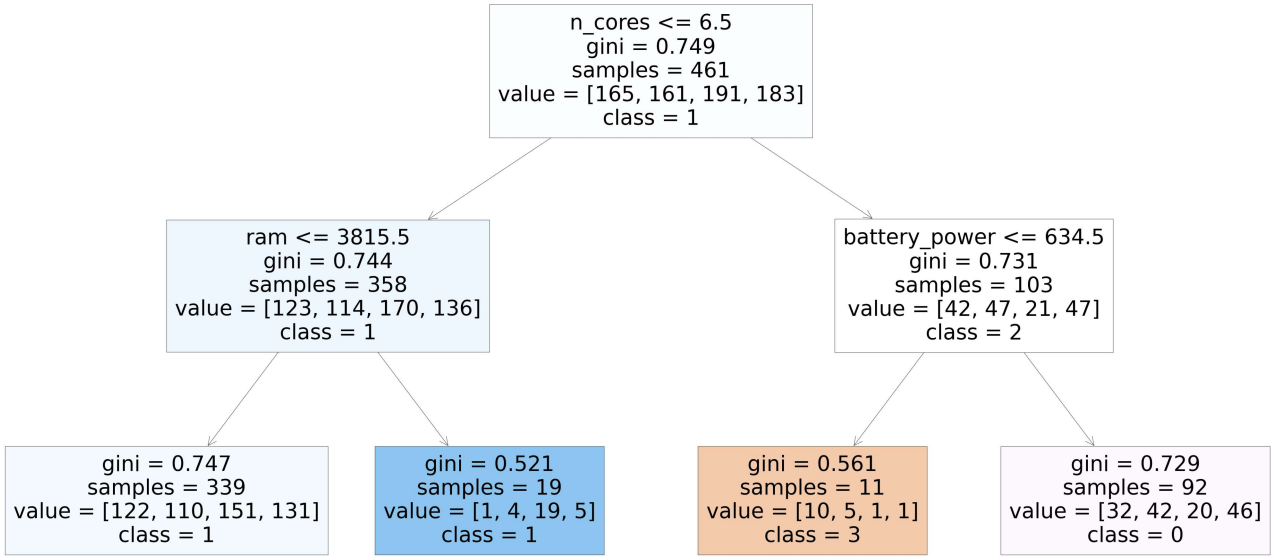
In [16]:
```
traindf['price_range'].value_counts()
```

Out[16]:
```
price_range
3    276
2    248
0    242
1    234
Name: count, dtype: int64
```

In [17]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[4],feature_names=X.columns,class_names=['3','2','1','0'],filled=True);
```




In [18]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[5],feature_names=X.columns,class_names=['3','2','1','0'],filled=True);
```



In [19]:
```
1 rf_best.feature_importances_
```

Out[19]:
```
array([0.09413381, 0.09145702, 0.        , 0.03081771, 0.        ,
       0.02576675, 0.02119797, 0.01673645, 0.        , 0.09333031,
       0.14072194, 0.0270046 , 0.1416834 , 0.02721845, 0.13436328,
       0.        , 0.        , 0.07854065, 0.        , 0.05156172,
       0.02546597])
```

In [20]:
```
1 imp_df=pd.DataFrame({"Varname":X_train.columns,"Imp":rf_best.feature_importances_})
```

In [21]:
```
1 imp_df.sort_values(by="Imp",ascending=False)
```

Out[21]:

| | Varname | Imp |
|---|---|---|
| 12 | px_height | 0.141683 |
| 10 | n_cores | 0.140722 |
| 14 | ram | 0.134363 |
| 0 | id | 0.094134 |
| 9 | mobile_wt | 0.093330 |
| 1 | battery_power | 0.091457 |
| 17 | talk_time | 0.078541 |
| 19 | touch_screen | 0.051562 |
| 3 | clock_speed | 0.030818 |
| 13 | px_width | 0.027218 |
| 11 | pc | 0.027005 |
| 5 | fc | 0.025767 |
| 20 | wifi | 0.025466 |
| 6 | four_g | 0.021198 |
| 7 | int_memory | 0.016736 |
| 8 | m_dep | 0.000000 |
| 2 | blue | 0.000000 |
| 15 | sc_h | 0.000000 |
| 16 | sc_w | 0.000000 |
| 18 | three_g | 0.000000 |
| 4 | dual_sim | 0.000000 |

In [ ]:
```
1
```