In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt,seaborn as sns
```

In [3]:

```
train_df=pd.read_csv(r"C:\Users\raja\Downloads\Mobile_Price_Classification_train.csv")
train_df
```

Out[3]:

dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	th
0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	
1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	
1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9	
0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11	
0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1	0	1	2	0.8	106	6	...	1222	1890	668	13	4	19	
1	0	0	39	0.2	187	4	...	915	1965	2032	11	10	16	
1	1	1	36	0.7	108	8	...	868	1632	3057	9	1	5	
0	4	1	46	0.1	145	5	...	336	670	869	18	10	19	
1	5	1	45	0.9	168	6	...	483	754	3919	19	4	2	

In [4]:

```
test_df=pd.read_csv(r"C:\Users\raja\Downloads\Mobile_Price_Classification_test.csv")
test_df
```

Out[4]:

blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc_h	sc_w
1	1.8	1	14	0	5	0.1	193	...	16	226	1412	3476	12	7
1	0.5	1	4	1	61	0.8	191	...	12	746	857	3895	6	0
1	2.8	0	1	0	27	0.9	186	...	4	1270	1366	2396	17	10
0	0.5	1	18	1	25	0.5	96	...	20	295	1752	3893	10	0
0	1.4	0	11	1	49	0.5	108	...	18	749	810	1773	15	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	1.9	0	0	1	54	0.5	170	...	17	644	913	2121	14	8
0	1.8	1	0	0	13	0.9	186	...	2	1152	1632	1933	8	1
0	1.4	0	1	1	8	0.5	80	...	12	477	825	1223	5	0
1	0.5	1	0	0	50	0.4	171	...	12	38	832	2509	15	11
1	0.5	0	4	1	35	0.1	140	...	19	457	608	2828	9	2

In [9]:

```
x=train_df.drop('price_range',axis=1)
y=train_df['price_range']
```

In [10]:

```
train_df['four_g'].value_counts()
```

Out[10]:

```
four_g
1    1043
0     957
Name: count, dtype: int64
```

In [11]:

```
x=train_df.drop('price_range',axis=1)
y=train_df['price_range']
```

In [12]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.7,random_state=42)
x_train.shape,x_test.shape
```

Out[12]:

```
((1400, 20), (600, 20))
```

In [13]:

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[13]:

▸ RandomForestClassifier

In [14]:

```
rf=RandomForestClassifier()
```

In [15]:

```
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 [16]:

```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rf,param_grid=params,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[16]:

▸ GridSearchCV

▸ estimator: RandomForestClassifier

▸ RandomForestClassifier

In [17]:

```
grid_search.best_score_
```

Out[17]:

```
0.8328571428571429
```

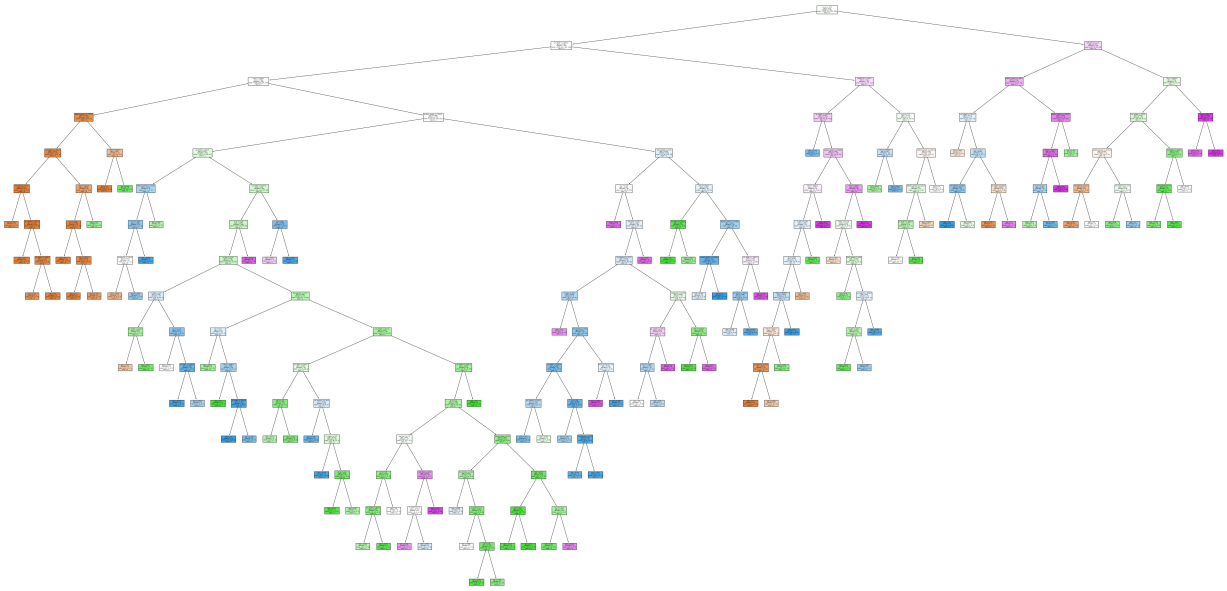
In [18]:

```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=20, min_samples_leaf=5)
```

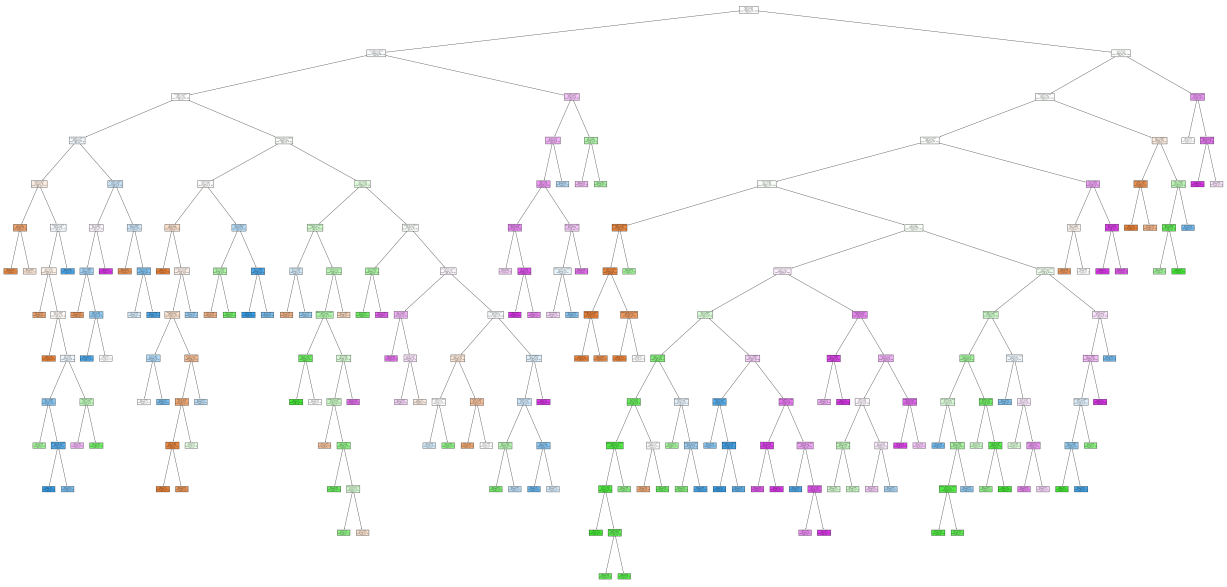
In [20]:

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



In [22]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[7],feature_names=x.columns,class_names=['0','1','2','3'],filled=True);
```



In [23]:

```
rf_best.feature_importances_
```

Out[23]:

```
array([0.07434881, 0.0043182 , 0.02066101, 0.00485052, 0.01704553,
       0.00509914, 0.03092368, 0.01694181, 0.02948496, 0.01380429,
       0.0186988 , 0.04885466, 0.05037508, 0.58918237, 0.01915089,
       0.02033392, 0.0214917 , 0.00395349, 0.00487837, 0.00560278])
```

In [24]:

```
imp_df=pd.DataFrame({"Varname":x_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[24]:

	Varname	Imp
13	ram	0.589182
0	battery_power	0.074349
12	px_width	0.050375
11	px_height	0.048855
6	int_memory	0.030924
8	mobile_wt	0.029485
16	talk_time	0.021492
2	clock_speed	0.020661
15	sc_w	0.020334
14	sc_h	0.019151
10	pc	0.018699
4	fc	0.017046
7	m_dep	0.016942
9	n_cores	0.013804
19	wifi	0.005603
5	four_g	0.005099
18	touch_screen	0.004878
3	dual_sim	0.004851
1	blue	0.004318
17	three_g	0.003953

In [25]:

```
x=test_df.drop('wifi',axis=1)
y=test_df['wifi']
```

In [26]:

```
test_df['four_g'].value_counts()
```

Out[26]:

```
four_g
0    513
1    487
Name: count, dtype: int64
```

In [27]:

```
x=test_df.drop('wifi',axis=1)
y=test_df['wifi']
```

In [28]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.7,random_state=42)
x_train.shape,x_test.shape
```

Out[28]:

```
((700, 20), (300, 20))
```

In [29]:

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_test,y_test)
```

Out[29]:

```
RandomForestClassifier
RandomForestClassifier()
```

In [30]:

```
rf=RandomForestClassifier()
```

In [31]:

```
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 [35]:

```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rf,param_grid=params,cv=2,scoring="accuracy")
grid_search.fit(x_test,y_test)
```

Out[35]:

```
GridSearchCV
estimator: RandomForestClassifier
RandomForestClassifier
```

In [33]:

```
grid_search.best_score_
```

Out[33]:

```
0.5366666666666666
```

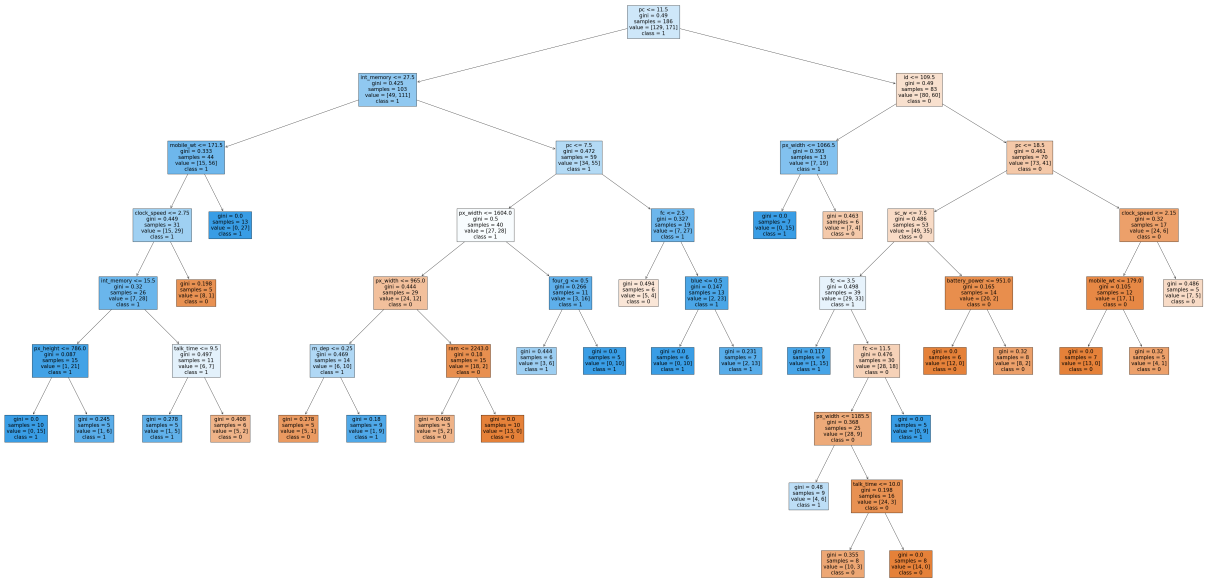
In [36]:

```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=20, min_samples_leaf=5, n_estimators=10)
```

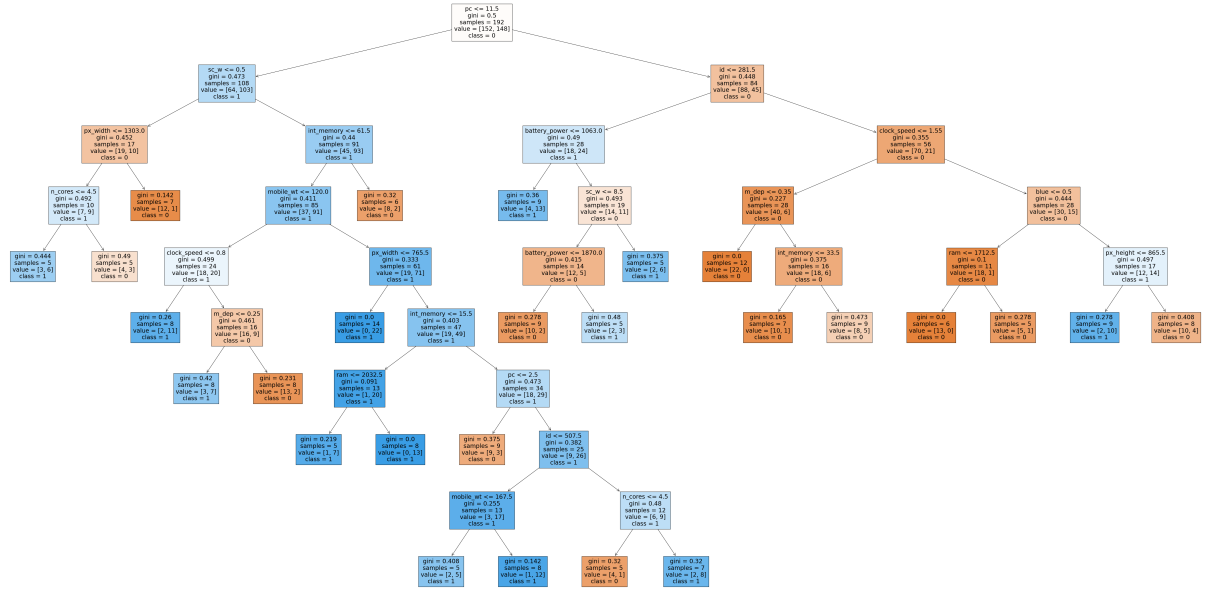
In [37]:

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



In [38]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[7],feature_names=x.columns,class_names=['0','1','2','3'],filled=True);
```



In [39]:

```
rf_best.feature_importances_
```

Out[39]:

```
array([0.11001855, 0.03613996, 0.03454918, 0.05752844, 0.00645731,
       0.04716106, 0.01412928, 0.04868597, 0.04308591, 0.07856564,
       0.02822526, 0.07850069, 0.06799408, 0.12203989, 0.08652807,
       0.04214072, 0.04591817, 0.04619242, 0.        , 0.0061394 ])
```

In [41]:

```
imp_df=pd.DataFrame({"Varname":x_train.columns,"Imp":rf_best.feature_importances_})
imp_df.sort_values(by="Imp",ascending=False)
```

Out[41]:

	Varname	Imp
13	px_width	0.122040
0	id	0.110019
14	ram	0.086528
9	mobile_wt	0.078566
11	pc	0.078501
12	px_height	0.067994
3	clock_speed	0.057528
7	int_memory	0.048686
5	fc	0.047161
17	talk_time	0.046192
16	sc_w	0.045918
8	m_dep	0.043086
15	sc_h	0.042141
1	battery_power	0.036140
2	blue	0.034549
10	n_cores	0.028225
6	four_g	0.014129
4	dual_sim	0.006457
19	touch_screen	0.006139
18	three_g	0.000000