In [1]:

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

In [2]:

```
df=pd.read_csv('health.csv')
df
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	O
0	6	148	72	35	0	33.6	0.627	50	
1	1	85	66	29	0	26.6	0.351	31	
2	8	183	64	0	0	23.3	0.672	32	
3	1	89	66	23	94	28.1	0.167	21	
4	0	137	40	35	168	43.1	2.288	33	
...	...	...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	0.171	63	
764	2	122	70	27	0	36.8	0.340	27	
765	5	121	72	23	112	26.2	0.245	30	
766	1	126	60	0	0	30.1	0.349	47	

In [3]:

```
df.columns
```

Out[3]:

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insuli
n',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [9]:

```
x=df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'Age']]
y=df['Outcome']
```

In [10]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.70)
```

In [11]:

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

Out[11]:

```
RandomForestClassifier()
```

Depth of Tree

In [12]:

```
parameters={"max_depth":[1,2,3,4,5],"min_samples_leaf":[5,23,45,76,78],'n_estimators':[10
```

Cross Validate

In [13]:

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

Out[13]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 23, 45, 76, 78],
                         'n_estimators': [10, 23, 45, 65, 7]},
             scoring='accuracy')
```

Score

In [14]:

```
grid_search.best_score_
```

Out[14]:

```
0.7260869565217392
```

In [15]:

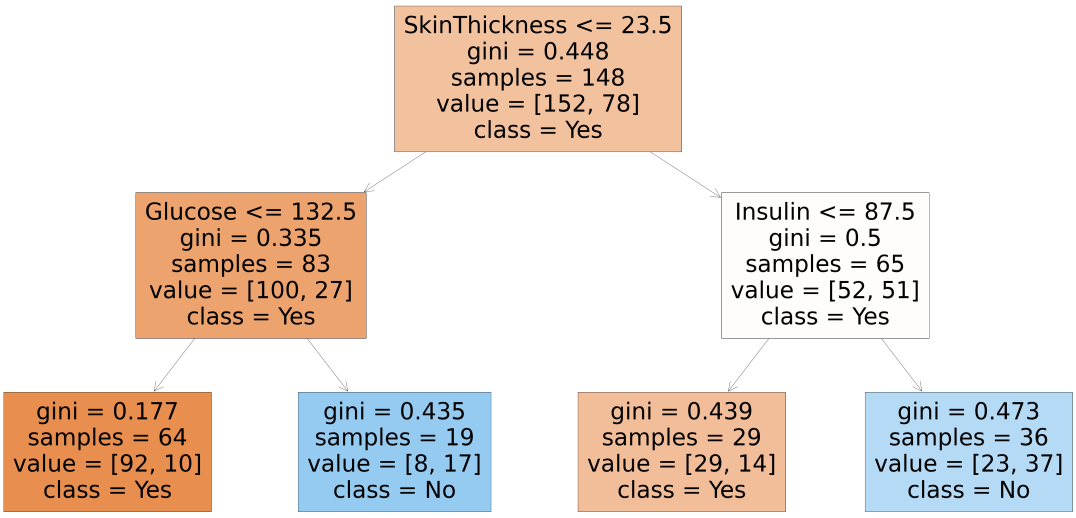
```
rfc_best=grid_search.best_estimator_
```

In [16]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled
```

Out[16]:

```
[Text(2232.0, 1812.0, 'SkinThickness <= 23.5\ngini = 0.448\nsamples = 148
\nvalue = [152, 78]\nclass = Yes'),
 Text(1116.0, 1087.2, 'Glucose <= 132.5\ngini = 0.335\nsamples = 83\nvalue
= [100, 27]\nclass = Yes'),
 Text(558.0, 362.39999999999986, 'gini = 0.177\nsamples = 64\nvalue = [92,
10]\nclass = Yes'),
 Text(1674.0, 362.39999999999986, 'gini = 0.435\nsamples = 19\nvalue = [8,
17]\nclass = No'),
 Text(3348.0, 1087.2, 'Insulin <= 87.5\ngini = 0.5\nsamples = 65\nvalue =
[52, 51]\nclass = Yes'),
 Text(2790.0, 362.39999999999986, 'gini = 0.439\nsamples = 29\nvalue = [2
9, 14]\nclass = Yes'),
 Text(3906.0, 362.39999999999986, 'gini = 0.473\nsamples = 36\nvalue = [2
3, 37]\nclass = No')]
```



In [ ]: