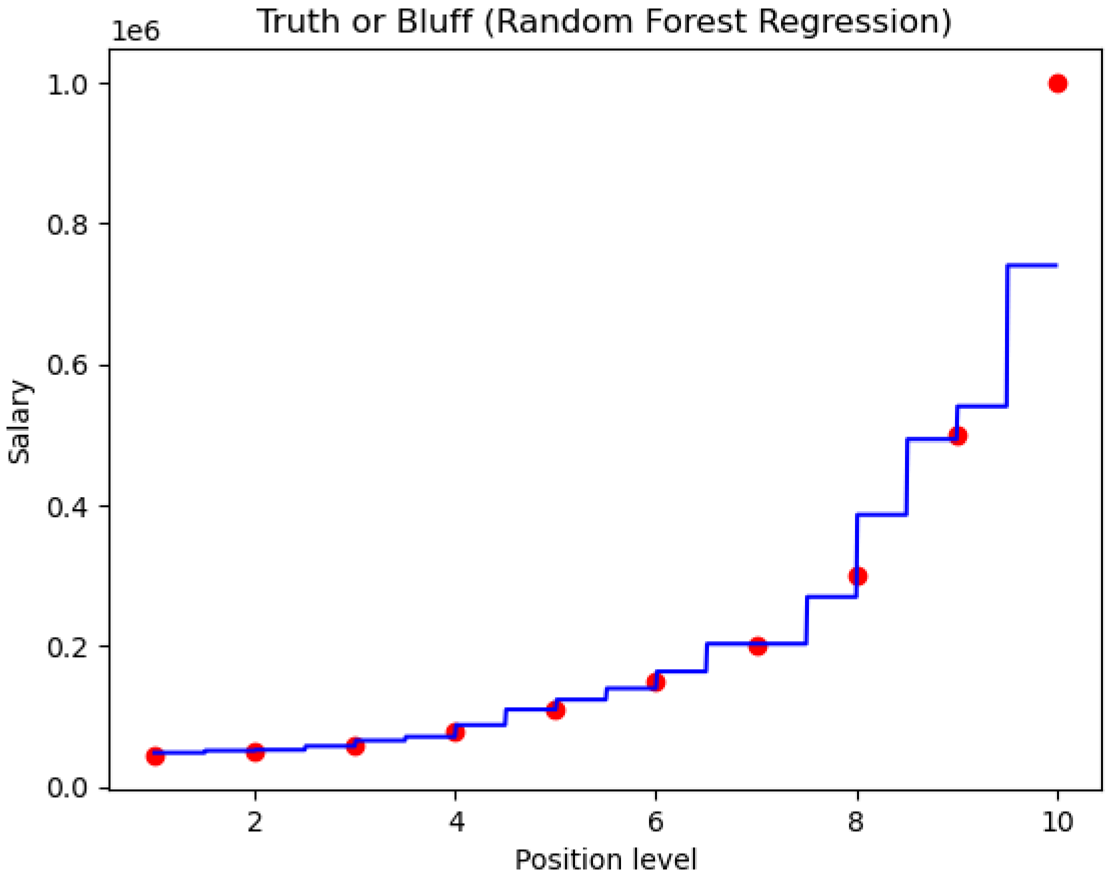
In [3]:

```
# Random Forest Regression

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
#dataset = pd.read_csv(r"C:\Users\kdata\Desktop\KODI WORK\1. NARESH\1. MORNING B
dataset = pd.read_csv(r'C:\Users\soham\OneDrive\Desktop\(OCT 02)\1.POLYNOMIAL RE
X = dataset.iloc[:, 1:2].values
y = dataset.iloc[:, 2].values

# Splitting the dataset into the Training set and Test set
"""from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, rando

'''
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)
'''

# Fitting Random Forest Regression to the dataset
#from sklearn.ensemble import RandomForestRegressor
#regressor = RandomForestRegressor(n_estimators = 100, random_state = 0)
#regressor.fit(X, y)

from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators=15)
regressor.fit(X, y)

# Predicting a new result
y_pred = regressor.predict([[6.5]])

# Visualising the Random Forest Regression results (higher resolution)one
X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'red')
plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
plt.title('Truth or Bluff (Random Forest Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

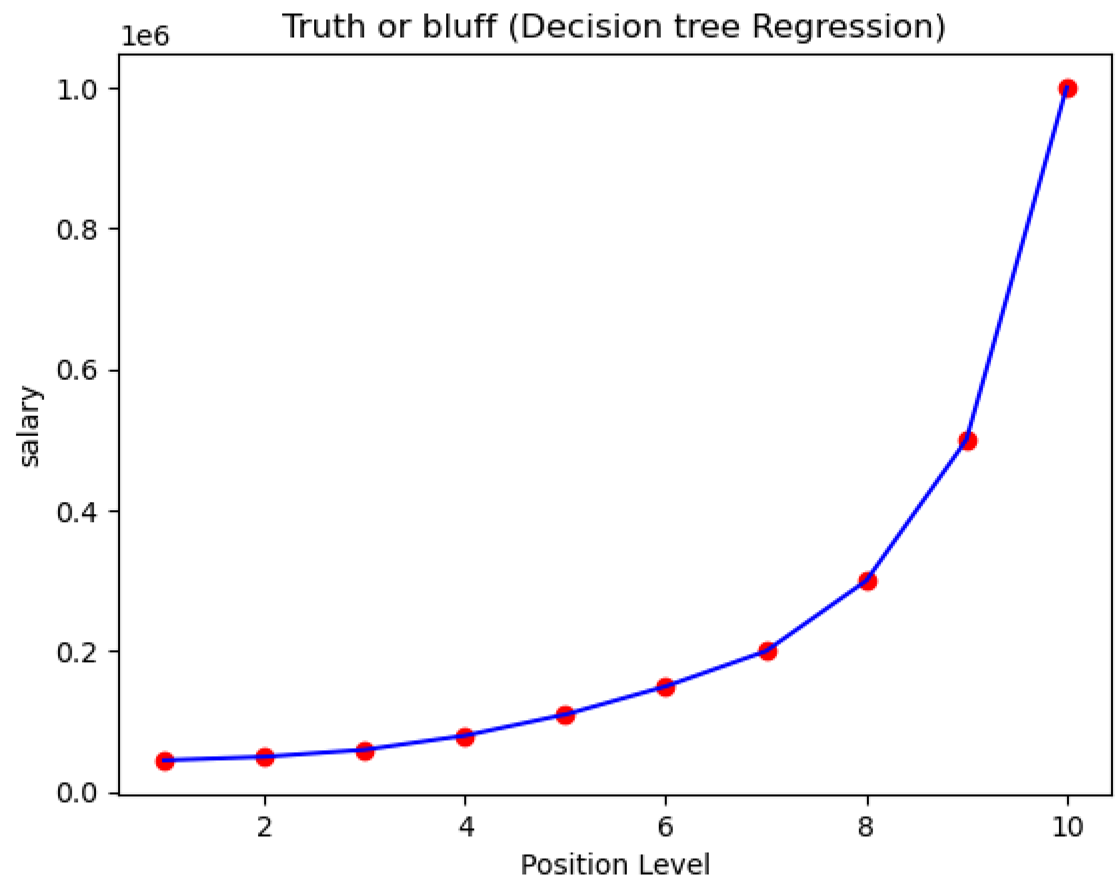
```
C:\Users\soham\AppData\Local\Temp\ipykernel_8324\1760639366.py:41: DeprecationWar
ning: Conversion of an array with ndim > 0 to a scalar is deprecated, and will er
ror in future. Ensure you extract a single element from your array before perform
ing this operation. (Deprecated NumPy 1.25.)
  X_grid = np.arange(min(X), max(X), 0.01)
```



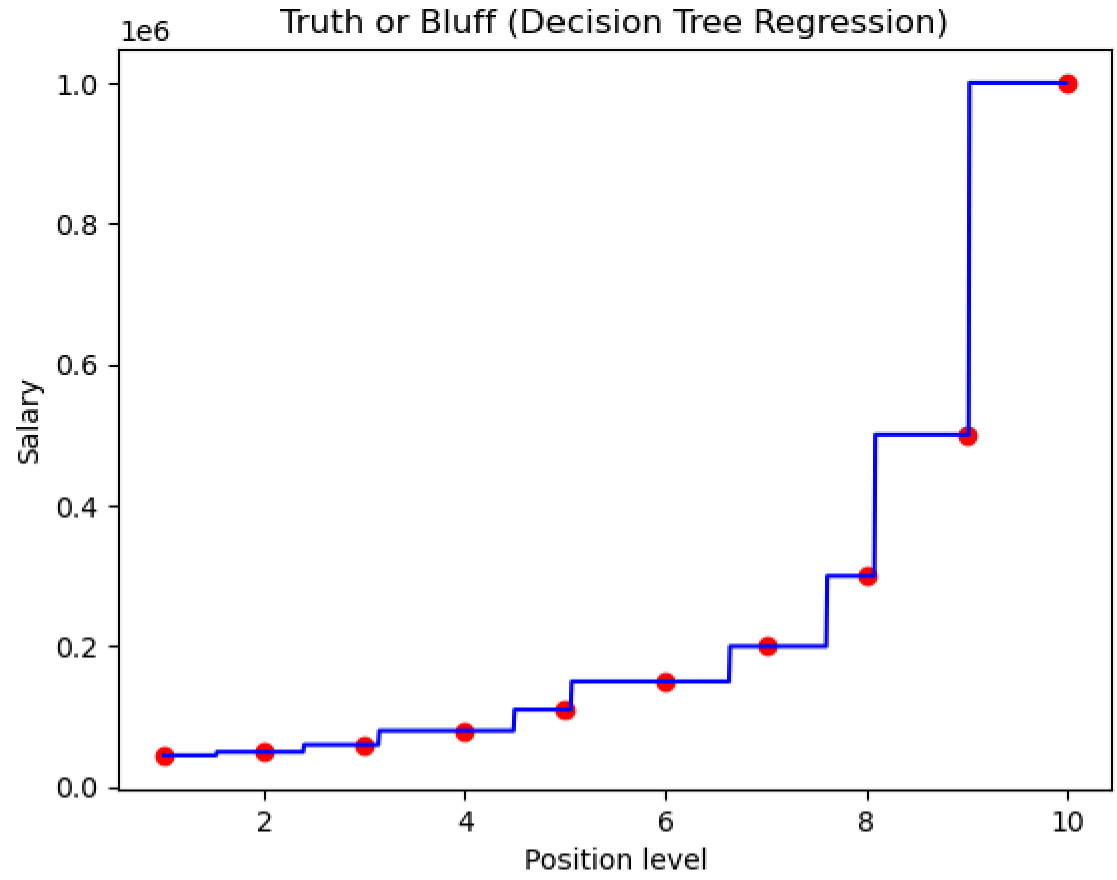
In [5]:

```
# Decision Tree Regression

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv(r'C:\Users\soham\OneDrive\Desktop\(OCT 02)\1.POLYNOMIAL RE
X = dataset.iloc[:, 1:2].values
y = dataset.iloc[:, 2].values

# Splitting the dataset into the Training set and Test set
"""from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, rando

'''# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)'''

#Fitting Decision Tree Regression to the datasetmae

from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(criterion = 'friedman_mse',splitter = 'random'
regressor.fit(X, y)

from sklearn.ensemble import RandomForestRegressor
reg = RandomForestRegressor(n_estimators = 300, random_state = 0)
```

```
reg.fit(X,y)

# Predicting a new result
y_pred = reg.predict([[6.5]])
#now predict previous employee salary & visualize the result
#emplyoee said his salary was 161k but as per dt we got as 150 which was sama as
#what we got in decision tree as 10k less as from previous salary



plt.scatter(X, y, color = 'red')
plt.plot(X,regressor.predict(X), color = 'blue')
plt.title('Truth or bluff (Decision tree Regression)')
plt.xlabel('Position Level')
plt.ylabel('salary')
plt.show()
#first part is curve is very good & as i explained this is not a decision tree c
#algorithm of decission tree is by considering the entrophy and information gain
#as per our tutorial we have 2 independent variable diferent interval forms rect
#you have quastion if you taking average of each interval then how do you have a
#And you cannot find the average of independent variable & this is not a continu
#lets plot the higher resolution using tree models


#if you advance visualisation along with tree structure then you will get this r
# Visualising the Decision Tree Regression results (higher resolution)
X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'red')
plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
plt.title('Truth or Bluff (Decision Tree Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()

#if you check the plot you found the straight & verticle line hear and based on
#if you check the interval of 6 then you get the point of 150k & the range is 5.
#this is all about decission tree regression & for next session we will see the
```



```
C:\Users\soham\AppData\Local\Temp\ipykernel_8324\1883255396.py:59: DeprecationWarning: Conversion of an array with ndim > 0 to a scalar is deprecated, and will error in future. Ensure you extract a single element from your array before performing this operation. (Deprecated NumPy 1.25.)
  X_grid = np.arange(min(X), max(X), 0.01)
```



In [ ]: