KARANJOT SINGH

In [182]:

```
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score
import warnings
warnings.filterwarnings("ignore")
```

In [183]:

```
iris = pd.read_csv(r'IRIS.csv')
iris.head()
```

Out[183]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [184]:

```
y = iris.species
x = iris.drop('species',axis = 1)
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state
= 100)
```

In [185]:

```
Dt = DecisionTreeClassifier()

Dt = Dt.fit(X_train,y_train)

predict_dt = Dt.predict(X_test)
ac_dt = accuracy_score(y_test,predict_dt)*100
print("Accuracy of Decision Tree is:",ac_dt)
#print(predict_dt)
```

```
Accuracy of Decision Tree is: 95.55555555555556
```

In [186]:

```
Rf = RandomForestClassifier()

Rf = Rf.fit(X_train,y_train)

predict_Rf = Rf.predict(X_test)
ac_rf = accuracy_score(y_test,predict_Rf)*100
print("Accuracy of Decision Tree is:",ac_rf)
#print(predict_dt)
```

```
Accuracy of Decision Tree is: 95.55555555555556
```

In [187]:

```
ab = AdaBoostClassifier()
ab.fit(X_train,y_train)

predict_ab = ab.predict(X_test)
ac_ab = accuracy_score(y_test,predict_ab)*100
print("Accuracy of Decision Tree is:",ac_ab)
```

```
Accuracy of Decision Tree is: 95.55555555555556
```

In [188]:

```
gb = GradientBoostingClassifier()

gb = gb.fit(X_train,y_train)

predict_gb = gb.predict(X_test)
ac_gb = accuracy_score(y_test,predict_gb)*100
print("Accuracy of Decision Tree is:",ac_gb)
#print(predict_dt)
```

```
Accuracy of Decision Tree is: 95.55555555555556
```

In [189]:

```
label = ["Decision Tree","Random Forest","Ada Boost","Gradient Boost"]
sns.barplot(label,[ac_dt,ac_rf,ac_ab,ac_gb])
```

Out[189]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ea4292e208>
```