

/Users/aqibaabdulqadir/P

Level 0:

5

Level 1:

5

3

7

Level 2:

5

3

2

4

7

8

Level 3:

5

3

2

4

8

7

Level 4:

5

3

2

4

8

7

```
graph = {
    '5': ['3', '7'],
    '3': ['2', '4'],
    '7': ['8'],
    '2': [],
    '4': ['8'],
    '8': []
}
```

}

```
def ids(start_node, max_depth):
```

```
    def dls(node, depth):
```

```
        if depth == 0:
```

```
            return [node]
```

```
        if node not in graph or depth < 0:
```

```
            return []
```

```
        result = [node]
```

```
        for neighbor in graph[node]:
```

```
            result.extend(dls(neighbor, depth - 1))
```

```
        return result
```

```
    for depth in range(max_depth + 1):
```

```
        print(f"Level {depth}:")
```

```
        visited_at_depth = set()
```

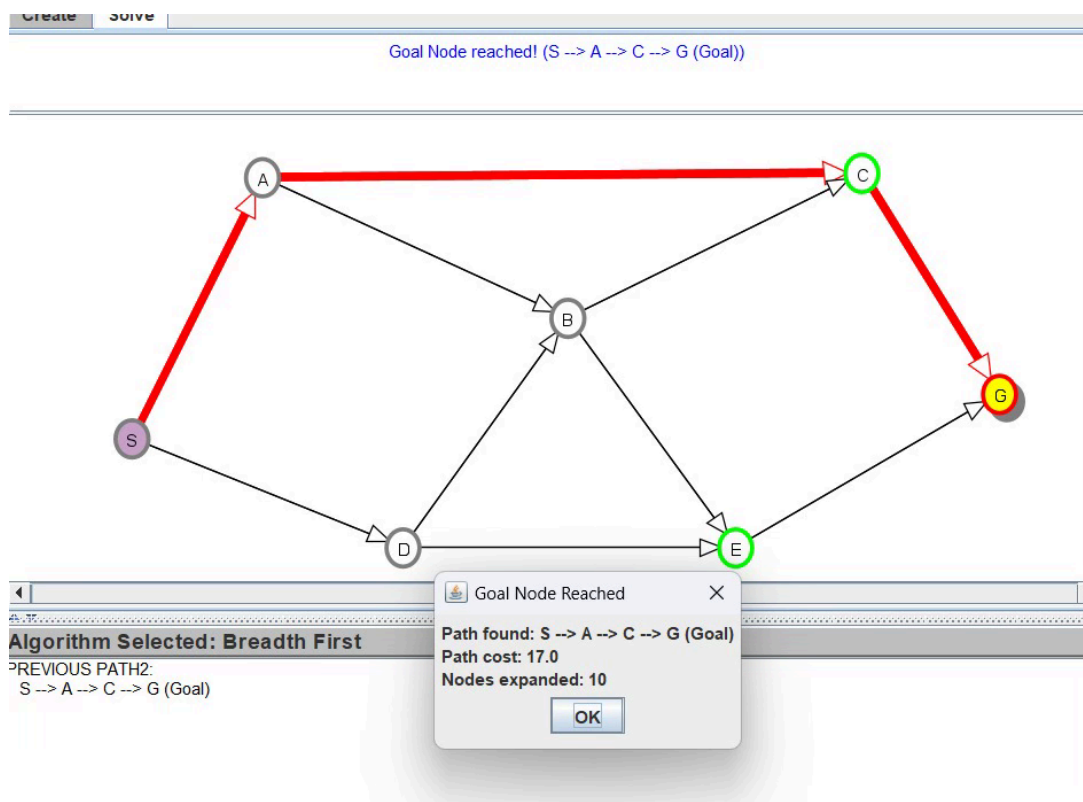
```
        for node in dls(start_node, depth):
```

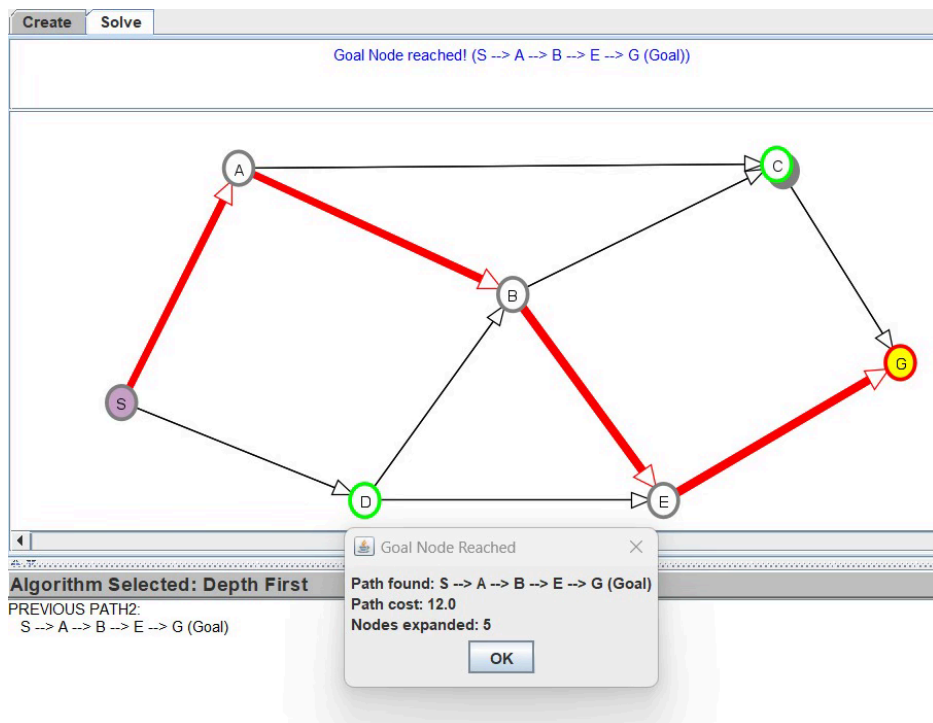
```
            if node not in visited_at_depth:
```

```
                print(node)
```

```
                visited_at_depth.add(node)
```

```
ids('5', 4)
```





heart

age	gender	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
40	1	1	100	260	0	1	174	0	0.0	0	0	0	1

heart-norm-train

age	gender	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1	1	1	0.481132075	0.244292237	1	0	0.603053435	0	0.370967742	0	0	0.333333333	1
2	1	0.666666667	0.339622642	0.283105023	0	0.5	0.885496183	0	0.564516129	0	0	0.666666667	1
3	0	0.333333333	0.339622642	0.178082192	0	0	0.770992366	0	0.225806452	1	0	0.666666667	1
4	1	0.333333333	0.245283019	0.251141553	0	0.5	0.816793893	0	0.129032258	1	0	0.666666667	1
5	0	0	0.245283019	0.520547945	0	0.5	0.702290076	1	0.096774194	1	0	0.666666667	1
6	1	0	0.433962264	0.150684932	0	0.5	0.58778626	0	0.064516129	0.5	0	0.333333333	1
7	0	0.333333333	0.433962264	0.383561644	0	0	0.625954198	0	0.209677419	0.5	0	0.666666667	1
8	1	0.333333333	0.245283019	0.312785388	0	0.5	0.778625954	0	0	1	0	1	1
9	1	0.666666667	0.735849057	0.166666667	1	0.5	0.694656489	0	0.080645161	1	0	1	1
10	1	0.666666667	0.528301887	0.095890411	0	0.5	0.786259542	0	0.258064516	1	0	0.666666667	1
11	1	0	0.433962264	0.257990868	0	0.5	0.679389313	0	0.193548387	1	0	0.666666667	1
12	0	0.666666667	0.339622642	0.340182648	0	0.5	0.519083969	0	0.032258065	1	0	0.666666667	1
13	1	0.333333333	0.339622642	0.319634703	0	0.5	0.763358779	0	0.096774194	1	0	0.666666667	1
14	1	1	0.150943396	0.194063927	0	0	0.557251908	1	0.290322581	0.5	0	0.666666667	1

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder, OrdinalEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
import tensorflow as tf
from tensorflow import keras

url = "heart.csv"
df = pd.read_csv(url)

numerical_features = ['age', 'chol', 'trestbps', 'thalach', 'oldpeak']
categorical_features = ['gender', 'cp', 'fbs', 'restecg', 'exang', 'ca', 'thal', 'slope']

target = 'target'

X = df.drop(columns=[target])
y = df[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

numerical_transformer = Pipeline(steps=[
    ('scaler', StandardScaler())
])

categorical_transformer = ColumnTransformer(transformers=[
    ('onehot', OneHotEncoder(), ['cp', 'thal', 'restecg']),
    ('ordinal', OrdinalEncoder(), ['slope'])
], remainder='passthrough')

preprocessor = ColumnTransformer(transformers=[
    ('num', numerical_transformer, numerical_features),
    ('cat', categorical_transformer, categorical_features)
])

model_pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor)
])

X_train_processed = model_pipeline.fit_transform(X_train)
X_test_processed = model_pipeline.transform(X_test)

X_train_df = pd.DataFrame(X_train_processed)
X_test_df = pd.DataFrame(X_test_processed)

input_shape = X_train_processed.shape[1]

model = keras.Sequential([
    keras.layers.InputLayer(shape=(input_shape,)),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

history = model.fit(X_train_processed, y_train, epochs=20, batch_size=32, validation_data=(X_test_processed, y_test))
test_loss, test_accuracy = model.evaluate(X_test_processed, y_test)
print(f"Test Accuracy: {test_accuracy:.2f}")
new_data = np.array([X_test_processed[0]])
prediction = model.predict(new_data)
print(f"Predicted class for the new data: {'Heart Disease' if prediction > 0.5 else 'No Heart Disease'}")

```