```
!pip install -U pm4py
!pip install visualization
import pandas as pd
from pm4py.objects.conversion.log import converter as log_converter
from pm4py.objects.log.importer.xes import importer as xes_importer
from pm4py.objects.log.util import dataframe_utils

from pm4py.algo.discovery.inductive import algorithm as inductive_miner
from pm4py.algo.discovery.alpha import algorithm as alpha_miner
from pm4py.algo.discovery.heuristics import algorithm as heuristics_miner
from pm4py.algo.discovery.dfg import algorithm as dfg_discovery
from pm4py.visualization.dfg import visualizer as dfg_visualization

from sklearn.metrics import pairwise_distances_argmin

from pm4py.objects.conversion.log import converter as log_converter
from pm4py.algo.discovery.alpha import algorithm as alpha_miner
from pm4py.visualization.petri_net import visualizer as pn_visualizer
from pm4py.visualization.petri_net.util import performance_map
from pm4py.visualization.process_tree import visualizer as pt_visualizer
from pm4py.visualization.heuristics_net import visualizer as hn_visualizer
from pm4py.visualization.process_tree import visualizer as pt_visualizer

from pm4py.objects.conversion.process_tree import converter as pt_converter

from pm4py.objects.conversion.log import converter as log_converter
from pm4py.objects.log.importer.xes import importer as xes_importer

from pm4py.algo.discovery.alpha import algorithm as alpha_miner
from pm4py.algo.discovery.inductive import algorithm as inductive_miner
from pm4py.algo.discovery.heuristics import algorithm as heuristics_miner
from pm4py.algo.discovery.dfg import algorithm as dfg_discovery

from pm4py.visualization.process_tree import visualizer as pt_visualizer
from pm4py.visualization.heuristics_net import visualizer as hn_visualizer
from pm4py.visualization.dfg import visualizer as dfg_visualization
```

```
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.7->matplotlib->pm4py) (1.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: visualization in /usr/local/lib/python3.9/dist-packages (1.0.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages (from visualization) (1.22.4)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.9/dist-packages (from visualization) (3.7.1)
Requirement already satisfied: autolab-core in /usr/local/lib/python3.9/dist-packages (from visualization) (1.1.1)
Requirement already satisfied: trimesh[easy] in /usr/local/lib/python3.9/dist-packages (from visualization) (3.21.5)
Requirement already satisfied: pyrender in /usr/local/lib/python3.9/dist-packages (from visualization) (0.1.45)
Requirement already satisfied: imageio in /usr/local/lib/python3.9/dist-packages (from visualization) (2.25.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.9/dist-packages (from autolab-core->visualization) (1.2.2)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.9/dist-packages (from autolab-core->visualization) (0.19.3)
Requirement already satisfied: joblib in /usr/local/lib/python3.9/dist-packages (from autolab-core->visualization) (1.2.0)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.9/dist-packages (from autolab-core->visualization) (4.7.0.72)
Requirement already satisfied: ruamel.yaml in /usr/local/lib/python3.9/dist-packages (from autolab-core->visualization) (0.17.21)
Requirement already satisfied: Pillow in /usr/local/lib/python3.9/dist-packages (from autolab-core->visualization) (8.4.0)
```

```
Requirement already satisfied: chardet in /usr/local/lib/python3.9/dist-packages (from trimesh[easy]->visualization) (4.0.0)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from trimesh[easy]->visualization) (2.27.1)
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/python3.9/dist-packages (from importlib-resources>=3.2.0->matplotlib->vi
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /usr/local/lib/python3.9/dist-packages (from jsonsche
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.9/dist-packages (from jsonschema->trimesh[easy]->visualization
Requirement already satisfied: dill>=0.3.6 in /usr/local/lib/python3.9/dist-packages (from multiprocess->autolab-core->visualization)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests->trimesh[easy]->visuali
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests->trimesh[easy]->visualizat
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests->trimesh[easy]->vis
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests->trimesh[easy]->visualization) (
Requirement already satisfied: ruamel.yaml.clib>=0.2.6 in /usr/local/lib/python3.9/dist-packages (from ruamel.yaml->autolab-core->vis
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.9/dist-packages (from scikit-image->autolab-core->visual
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.9/dist-packages (from scikit-image->autolab-core->visualiz
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from scikit-learn->autolab-core->visua
```

```
log = xes_importer.apply('running-example.xes')
```

```
parsing log, completed traces ::  6/6 [00:00<00:00,
100%
```

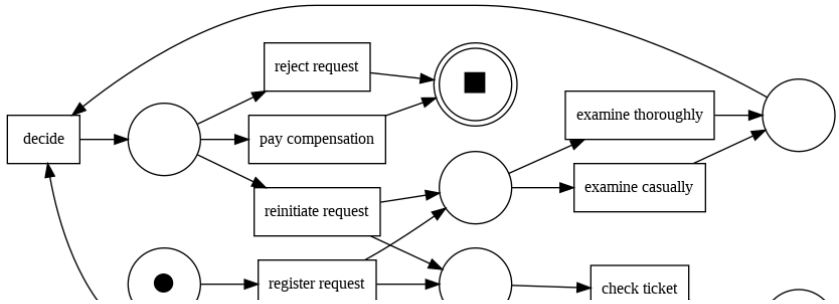
```
df = pd.read_csv('running-example.csv')
df = dataframe_utils.convert_timestamp_columns_in_df(df)
df = df.sort_values('time:timestamp')
log = log_converter.apply(df)
```

```
df.sort_values(['case:concept:name', 'time:timestamp']).reset_index(drop=True)
```

7	examine casually	400	Sean	2	Fluxicon Nitro	examine casually
8	decide	200	Sara	2	Fluxicon Nitro	decide
9	pay compensation	200	Ellen	2	Fluxicon Nitro	pay compensation
10	register request	50	Pete	3	Fluxicon Nitro	register request
11	examine casually	400	Mike	3	Fluxicon Nitro	examine casually
12	check ticket	100	Ellen	3	Fluxicon Nitro	check ticket
13	decide	200	Sara	3	Fluxicon Nitro	decide
14	reinitiate request	200	Sara	3	Fluxicon Nitro	reinitiate request
15	examine thoroughly	400	Sean	3	Fluxicon Nitro	examine thoroughly
16	check ticket	100	Pete	3	Fluxicon Nitro	check ticket
17	decide	200	Sara	3	Fluxicon Nitro	decide
18	pay compensation	200	Ellen	3	Fluxicon Nitro	pay compensation
19	register request	50	Pete	4	Fluxicon Nitro	register request
20	check ticket	100	Mike	4	Fluxicon Nitro	check ticket
21	examine thoroughly	400	Sean	4	Fluxicon Nitro	examine thoroughly
22	decide	200	Sara	4	Fluxicon Nitro	decide
23	reject request	200	Ellen	4	Fluxicon Nitro	reject request
24	register request	50	Ellen	5	Fluxicon Nitro	register request
25	examine casually	400	Mike	5	Fluxicon Nitro	examine casually
26	check ticket	100	Pete	5	Fluxicon Nitro	check ticket

```
net, initial_marking, final_marking = alpha_miner.apply(log)

gviz = pn_visualizer.apply(net, initial_marking, final_marking)
pn_visualizer.view(gviz)
```



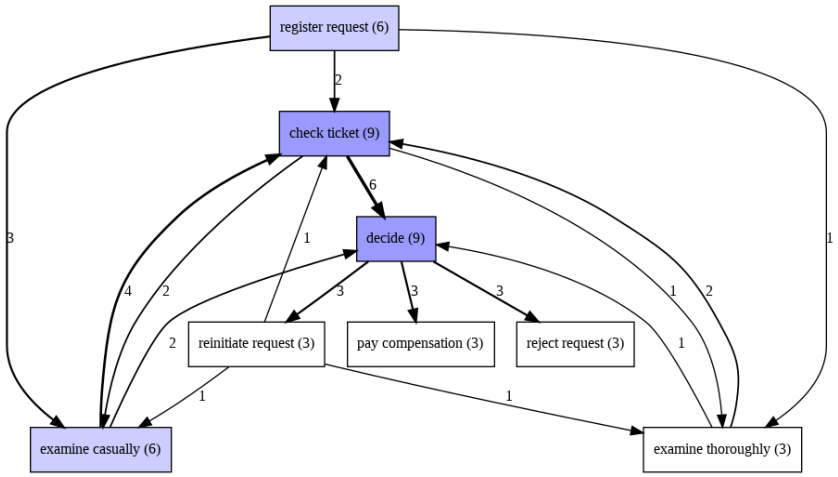

```
parameters = {pn_visualizer.Variants.FREQUENCY.value.Parameters.FORMAT: "png"}
gviz = pn_visualizer.apply(net, initial_marking, final_marking,
                           parameters=parameters,
                           variant=pn_visualizer.Variants.FREQUENCY,
                           log=log)
```

```
replaying log with TBR, completed                                  6/6 [00:00<00:00,
variants: 100%                                                     0.08it/s]
```

```
pn_visualizer.save(gviz, "alpha_miner_petri_net.png")
```

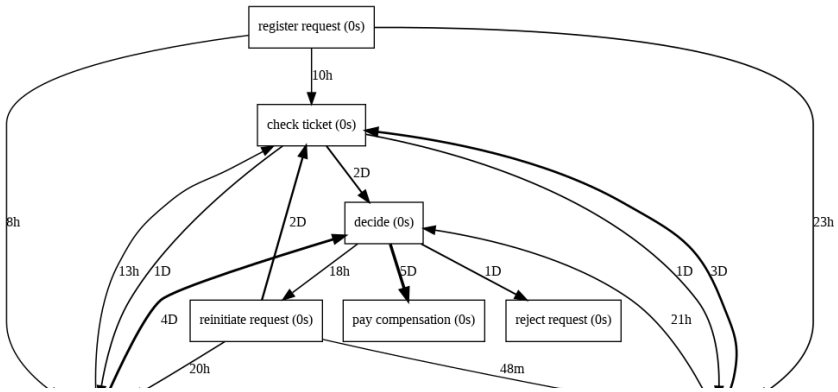
```
dfg = dfg_discovery.apply(log)
```

```
gviz = dfg_visualization.apply(dfg, log=log, variant=dfg_visualization.Variants.FREQUENCY)
dfg_visualization.view(gviz)
```



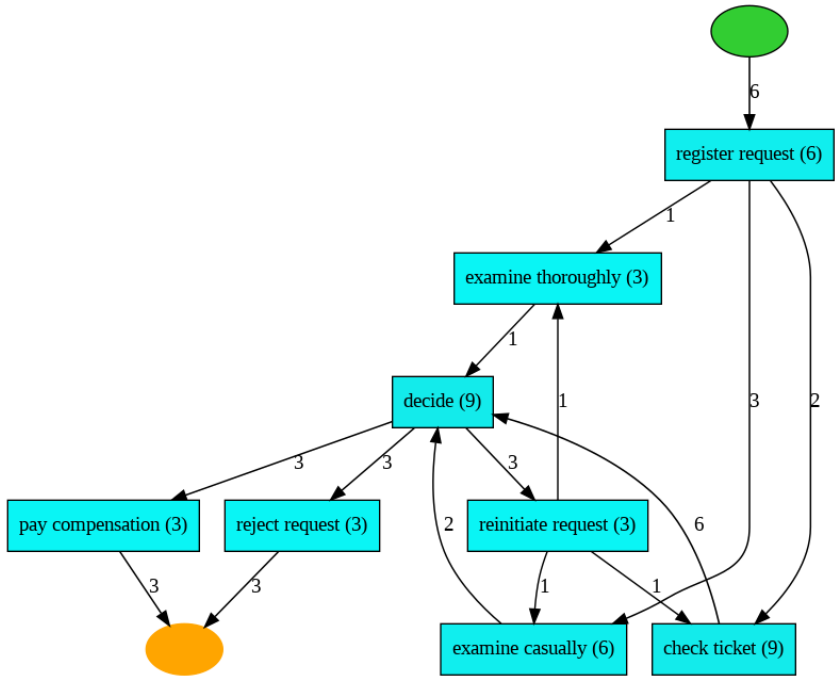

```
dfg = dfg_discovery.apply(log, variant=dfg_discovery.Variants.PERFORMANCE)
```

```
gviz = dfg_visualization.apply(dfg, log=log, variant=dfg_visualization.Variants.PERFORMANCE)
dfg_visualization.view(gviz)
```



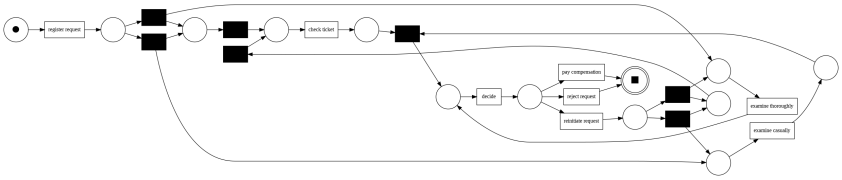

```
heu_net = heuristics_miner.apply_heu(log)
```

```
gviz = hn_visualizer.apply(heu_net)
hn_visualizer.view(gviz)
```



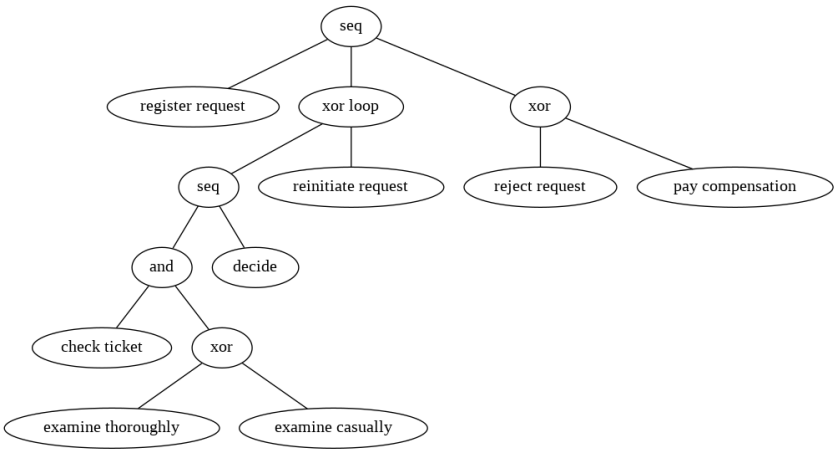

```
net, im, fm = heuristics_miner.apply(log)
```

```
gviz = pn_visualizer.apply(net, im, fm)
pn_visualizer.view(gviz)
```



```
tree = inductive_miner.apply(log)
```

```
gviz = pt_visualizer.apply(tree)
pt_visualizer.view(gviz)
```

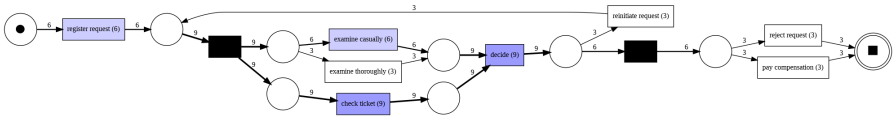



```
net, initial_marking, final_marking = pt_converter.apply(tree)

# alternatively, use the inductive_miner to create a petri net from scratch
# net, initial_marking, final_marking = inductive_miner.apply(log)

# viz
parameters = {pn_visualizer.Variants.FREQUENCY.value.Parameters.FORMAT: "png"}
gviz = pn_visualizer.apply(net, initial_marking, final_marking,
                           parameters=parameters,
                           variant=pn_visualizer.Variants.FREQUENCY,
                           log=log)
pn_visualizer.view(gviz)
```

```
replaying log with TBR, completed variants :: 100%  6/6 [00:00<00:00, 195.22it/s]
```



Double-click (or enter) to edit