

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.

```
!pip install -U pm4py
!pip install visualization
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pm4py
  Downloading pm4py-2.7.3-py3-none-any.whl (1.7 MB)
    1.7/1.7 MB 29.9 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages (from pm4py) (1.22.4)
Requirement already satisfied: pytz in /usr/local/lib/python3.9/dist-packages (from pm4py) (2022.7.1)
Requirement already satisfied: cvxopt in /usr/local/lib/python3.9/dist-packages (from pm4py) (1.3.0)
Collecting stringdist
  Downloading StringDist-1.0.9.tar.gz (7.4 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: lxml in /usr/local/lib/python3.9/dist-packages (from pm4py) (4.9.2)
Collecting intervaltree
  Downloading intervaltree-3.1.0.tar.gz (32 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: networkx in /usr/local/lib/python3.9/dist-packages (from pm4py) (3.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.9/dist-packages (from pm4py) (1.5.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.9/dist-packages (from pm4py) (3.7.1)
Requirement already satisfied: pydotplus in /usr/local/lib/python3.9/dist-packages (from pm4py) (2.0.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.9/dist-packages (from pm4py) (4.65.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.9/dist-packages (from pm4py) (1.10.1)
Collecting deprecation
  Downloading deprecation-2.1.0-py2.py3-none-any.whl (11 kB)
Requirement already satisfied: graphviz in /usr/local/lib/python3.9/dist-packages (from pm4py) (0.20.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-packages (from deprecation->pm4py) (23)
Requirement already satisfied: sortedcontainers<3.0,>=2.0 in /usr/local/lib/python3.9/dist-packages (from interv
Requirement already satisfied: importlib-resources>=3.2.0 in /usr/local/lib/python3.9/dist-packages (from matplo
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib->pm4py)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib->pm4
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib->pm4p
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib->pm4
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.9/dist-packages (from matplotlib->
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib->pm4p
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.9/dist-packages (from matplotlib->pm4py) (
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/python3.9/dist-packages (from importlib-resources)=
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.7->ma
Building wheels for collected packages: intervaltree, stringdist
  Building wheel for intervaltree (setup.py) ... done
  Created wheel for intervaltree: filename=intervaltree-3.1.0-py2.py3-none-any.whl size=26114 sha256=3b69b1fd12f
  Stored in directory: /root/.cache/pip/wheels/ab/fa/1b/75d9a713279796785711bd0bad8334aaace560c0bd28830c8c
  Building wheel for stringdist (setup.py) ... done
  Created wheel for stringdist: filename=StringDist-1.0.9-cp39-cp39-linux_x86_64.whl size=24702 sha256=d0426a67d
  Stored in directory: /root/.cache/pip/wheels/77/24/44/b12b9612fd07e141e6b4f1bb2001c621515bd0cac1014a783a
Successfully built intervaltree stringdist
Installing collected packages: stringdist, intervaltree, deprecation, pm4py
Successfully installed deprecation-2.1.0 intervaltree-3.1.0 pm4py-2.7.3 stringdist-1.0.9
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting visualization
  Downloading visualization-1.0.0-py3-none-any.whl (13 kB)
Collecting autolab-core
  Downloading autolab_core-1.1.1-py3-none-any.whl (116 kB)
    116.2/116.2 kB 4.6 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages (from visualization) (1.22.4)
Requirement already satisfied: imageio in /usr/local/lib/python3.9/dist-packages (from visualization) (2.25.1)
Collecting pyrender
  Downloading pyrender-0.1.45-py3-none-any.whl (1.2 MB)
    1.2/1.2 MB 37.4 MB/s eta 0:00:00
Requirement already satisfied: matplotlib in /usr/local/lib/python3.9/dist-packages (from visualization) (3.7.1)
```

```
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
```

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

```
/usr/local/lib/python3.9/dist-packages/pm4py/util/dt_parsing/parser.py:76: UserWarning:
  warnings.warn(
```

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

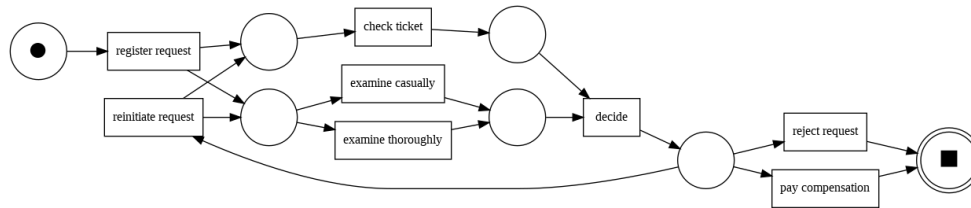
```
100% 119 74it/s]
```

```
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)
```

	Activity	Costs	Resource	case:concept:name	case:creator	concept:name
0	register request	50	Pete	1	Fluxicon Nitro	register request
1	examine thoroughly	400	Sue	1	Fluxicon Nitro	examine thoroughly
2	check ticket	100	Mike	1	Fluxicon Nitro	check ticket
3	decide	200	Sara	1	Fluxicon Nitro	decide
4	reject request	200	Pete	1	Fluxicon Nitro	reject request
5	register request	50	Mike	2	Fluxicon Nitro	register request
6	check ticket	100	Mike	2	Fluxicon Nitro	check ticket
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
	register					register

```
# alpha miner
net, initial_marking, final_marking = alpha_miner.apply(log)
# viz
gviz = pn_visualizer.apply(net, initial_marking, final_marking)
pn_visualizer.view(gviz)
```



```
# add information about frequency to the 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)

# save the Petri net
pn_visualizer.save(gviz, "alpha_miner_petri_net.png")
```

replaying log with TBR, completed

6/6 [00:00<00:00,

reinitiate 100%

151 704/1

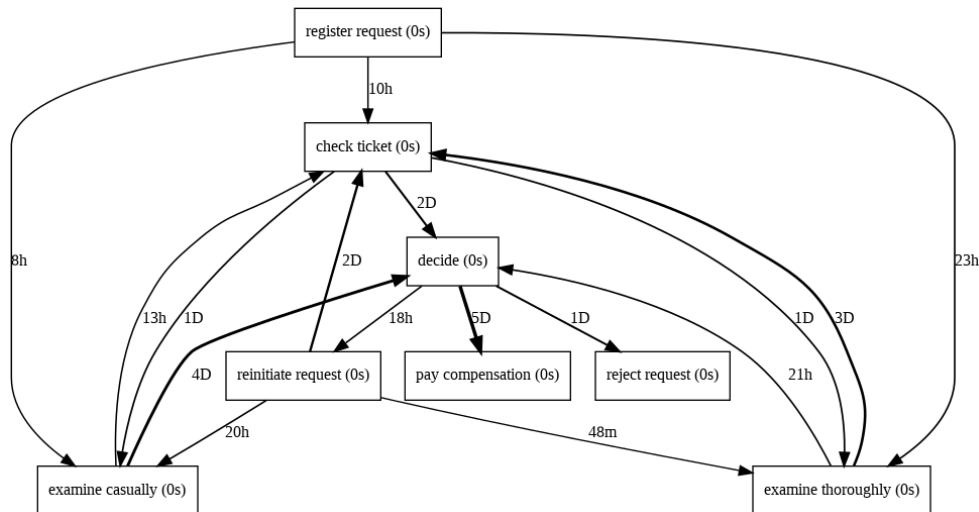
```
# creating the graph from log
dfg = dfg_discovery.apply(log)

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

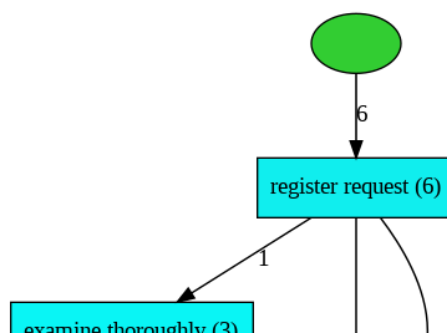
register request (6)

```
# creatig the graph from log
dfg = dfg_discovery.apply(log, variant=dfg_discovery.Variants.PERFORMANCE)

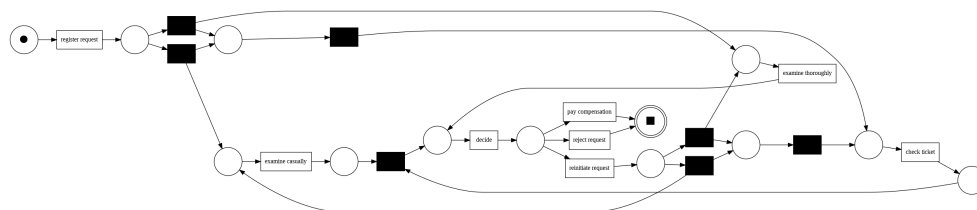
# viz
gviz = dfg_visualization.apply(dfg, log=log, variant=dfg_visualization.Variants.PERFORMANCE)
dfg_visualization.view(gviz)
```



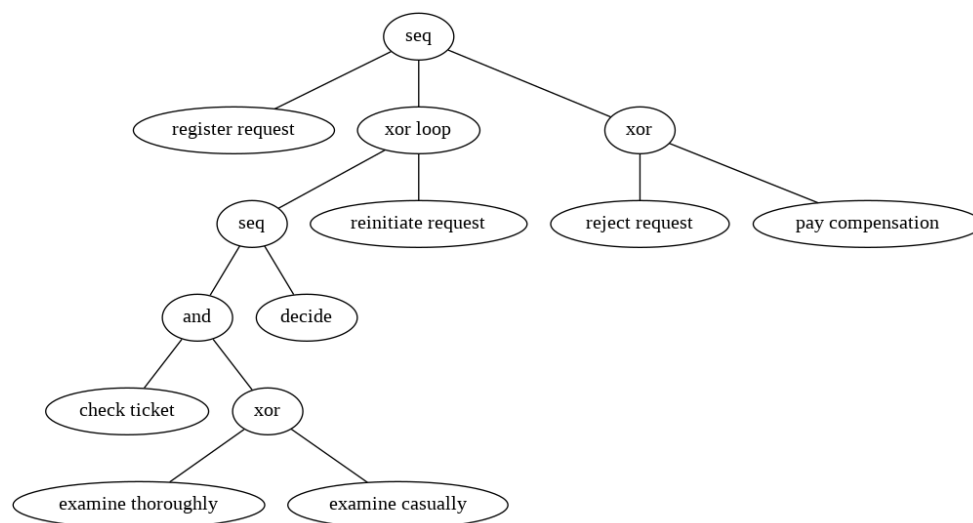
```
# heuristics miner
heu_net = heuristics_miner.apply_heu(log)
# viz
gviz = hn_visualizer.apply(heu_net)
hn_visualizer.view(gviz)
```



```
# heuristics miner
net, im, fm = heuristics_miner.apply(log)
# viz
gviz = pn_visualizer.apply(net, im, fm)
pn_visualizer.view(gviz)
```



```
# create the process tree
tree = inductive_miner.apply(log)
# viz
gviz = pt_visualizer.apply(tree)
pt_visualizer.view(gviz)
```



```
# convert the process tree to a petri net
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)

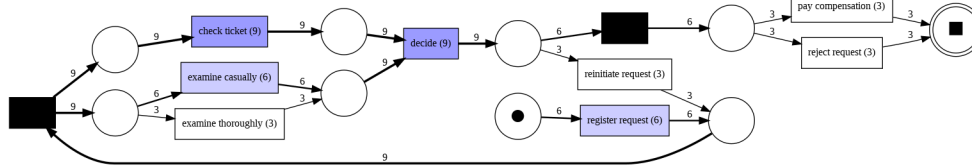
# viz
parameters = {pn_visualizer.Variants.FREQUENCY.value.Parameters.FORMAT: "png"}
gviz = pn_visualizer.apply(net, initial_marking, final_marking,
                           parameters=parameters,
                           variant=pn_visualizer.Variants.FREQUENCY,
                           log=log)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

replaying log with TBR, completed

variants :: 100%

6/6 [00:00<00:00,

116.24it/s]



✓ 0s completed at 10:25



Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.