Отчет

Евсеев Евгений Васильевич — БПИ212

Выводы по результатам тестов

In [ ]:

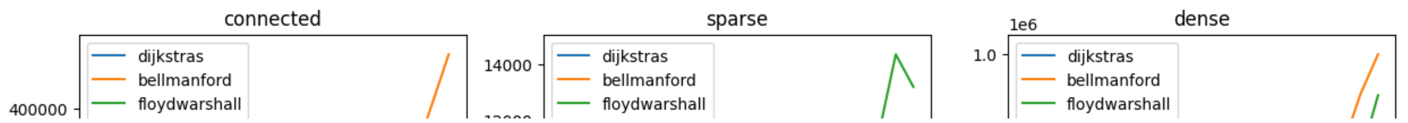
```
import matplotlib.pyplot as plt
import pandas as pd
```

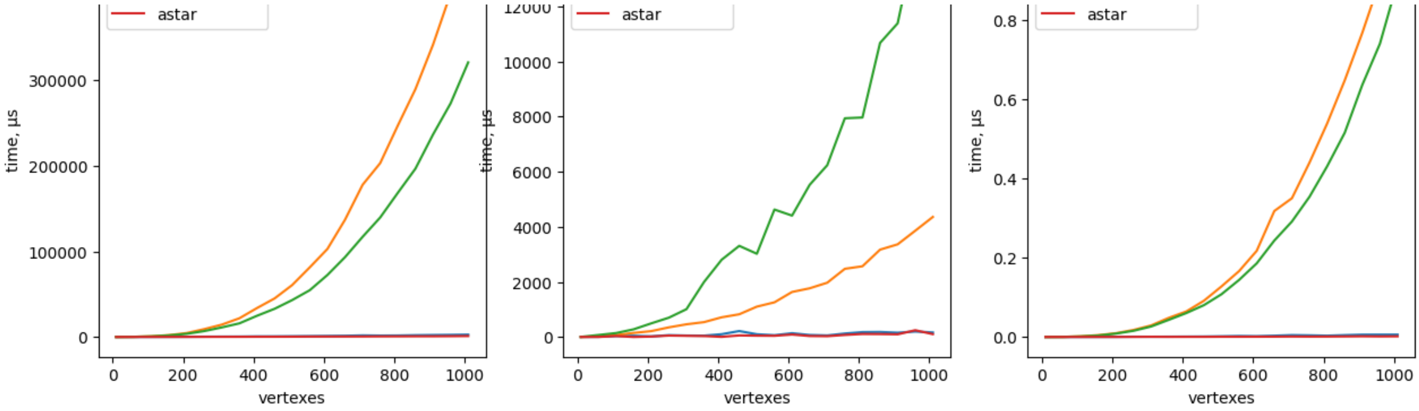
In [336]:

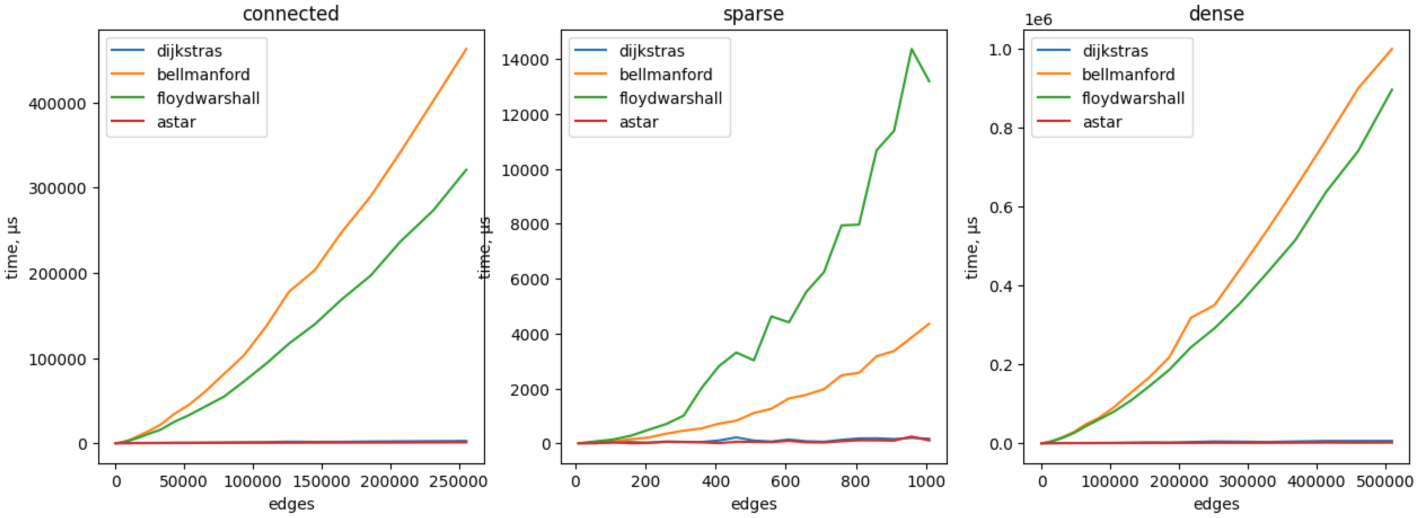
```
ALGOS = ['dijkstras', 'bellmanford', 'floydwarshall', 'astar']

connected = pd.read_csv('../tests_results/Connected_graph.csv', sep=';').set_index('algo'
).T
sparse = pd.read_csv('../tests_results/Sparse_graph.csv', sep=';').set_index('algo').T
dense = pd.read_csv('../tests_results/Dense_graph.csv', sep=';').set_index('algo').T

data = {
    'connected': connected,
    'sparse': sparse,
    'dense': dense
}

for df in data.values():
    df.columns = df.columns.map(lambda x: x.lower())
    df['vertexes'] = df.index.map(lambda x: int(x.split('@')[0]))
    df['edges'] = df.index.map(lambda x: int(x.split('@')[1]))
    df.set_index(['vertexes', 'edges'], inplace=True)
    df.columns.name = None
```

In [337]:

```
# plot by vertexes
for ind, (name, df) in enumerate(data.items()):
    df_by_vertexes = df.copy()
    df_by_vertexes.index = df_by_vertexes.index.map(lambda x: x[0])
    df_by_vertexes.index.name = 'vertexes'

    pl = df_by_vertexes.plot(title=name, ax=plt.subplot(1, len(data), ind + 1), figsize=
(15, 5))
    pl.set_xlabel('vertexes')
    pl.set_ylabel('time, µs')

plt.show()

# plot by edges
for ind, (name, df) in enumerate(data.items()):
    df_by_edges = df.copy()
    df_by_edges.index = df_by_edges.index.map(lambda x: x[1])
    df_by_edges.index.name = 'edges'

    pl = df_by_edges.plot(title=name, ax=plt.subplot(1, len(data), ind + 1), figsize=(15
, 5))
    pl.set_xlabel('edges')
    pl.set_ylabel('time, µs')

plt.show()
```

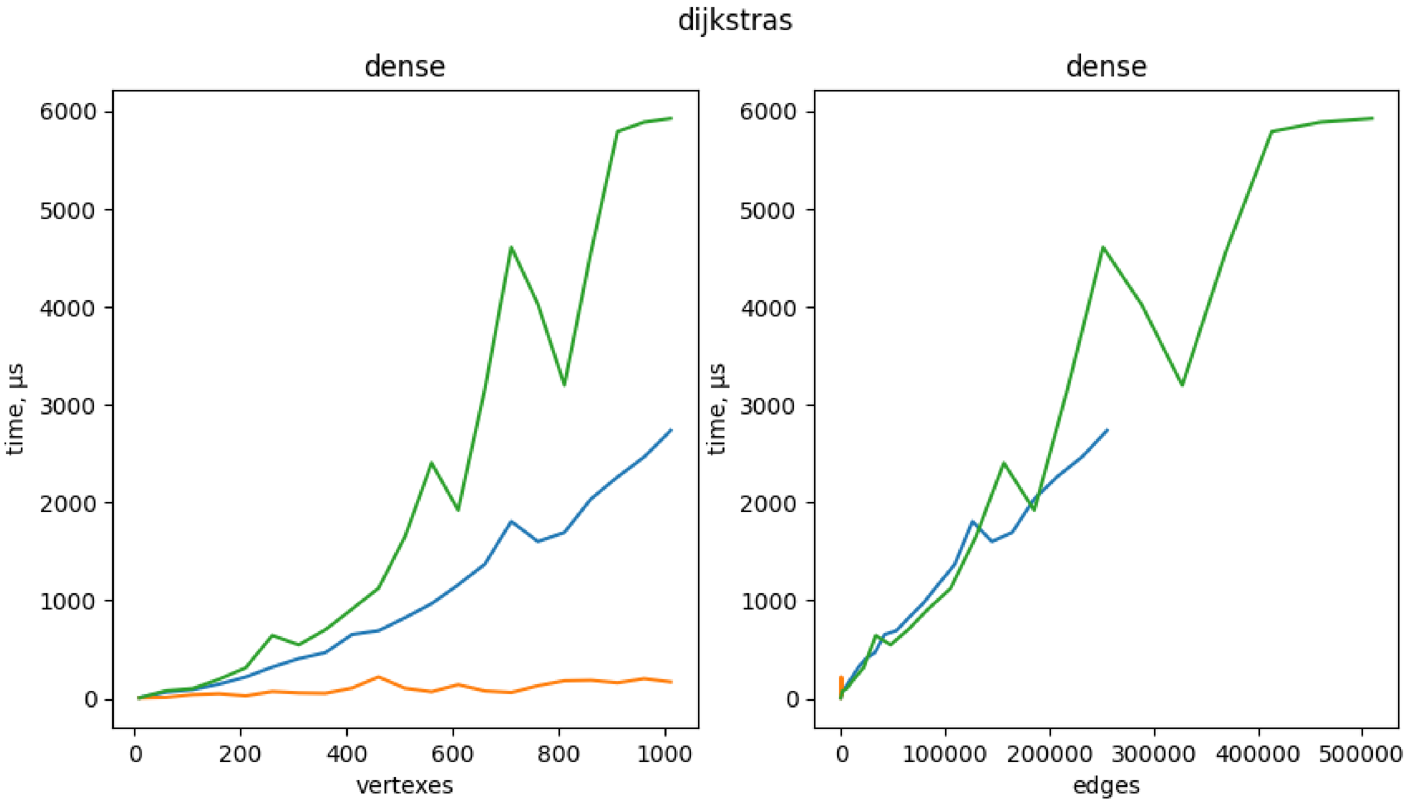


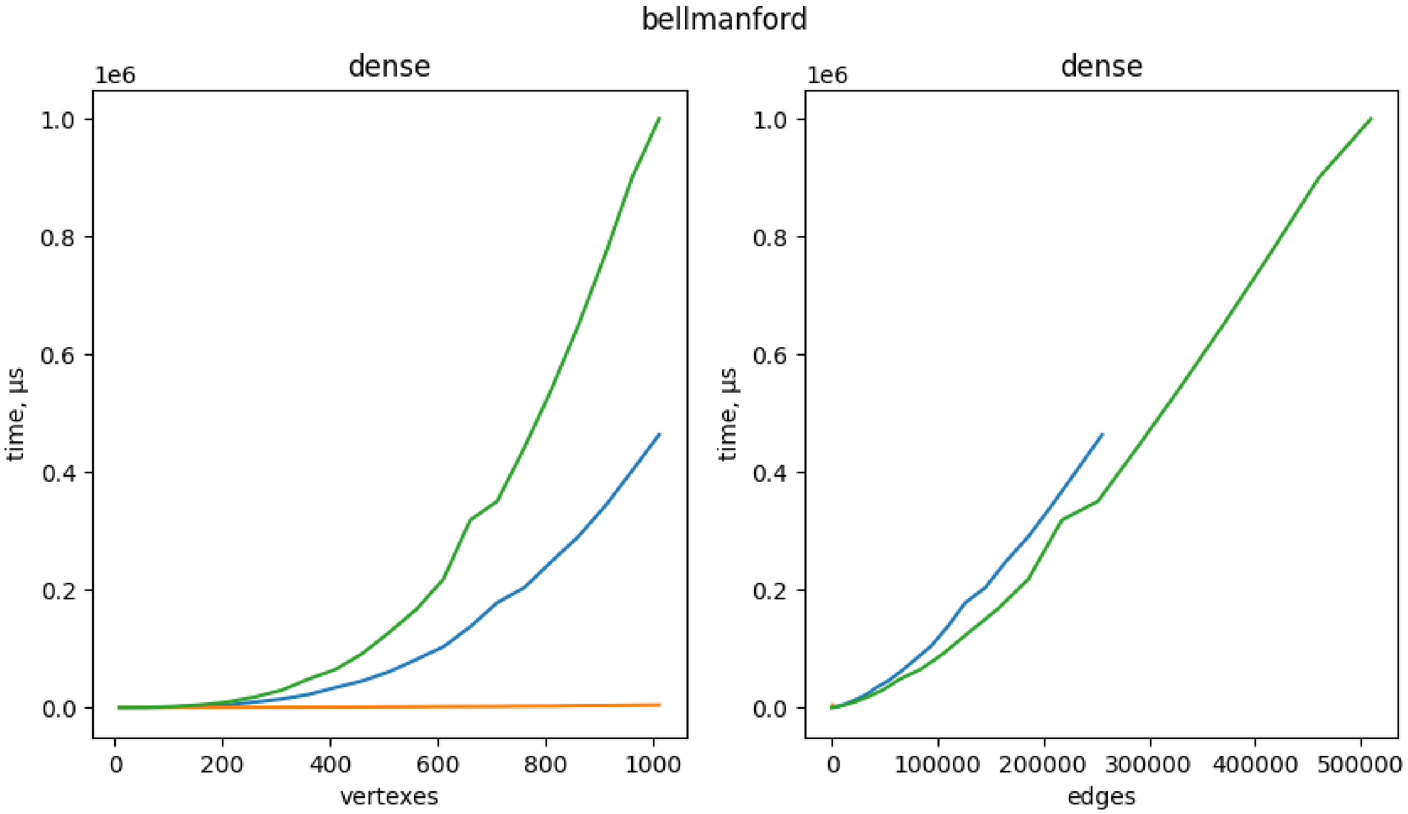


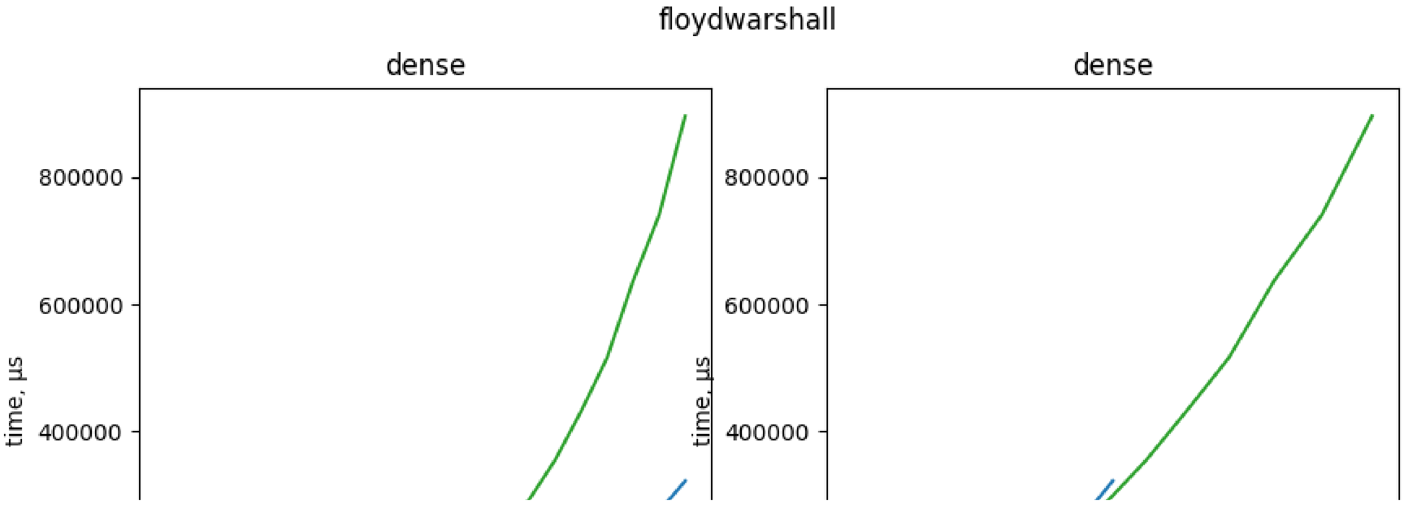


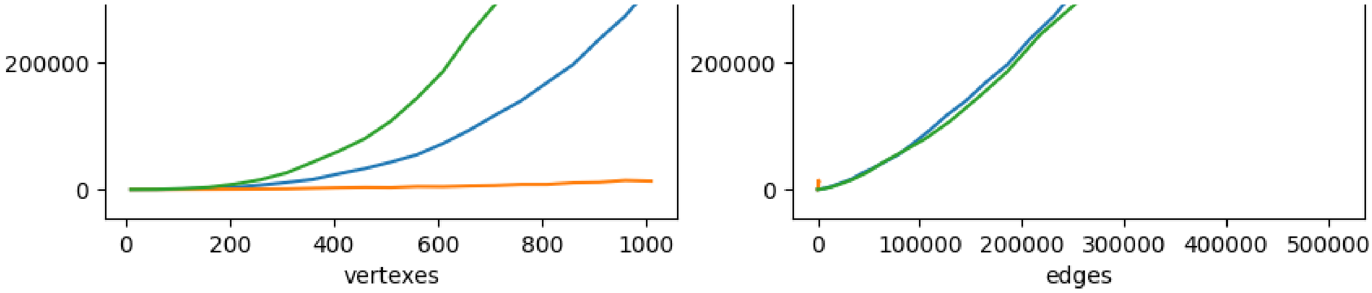
In [338]:

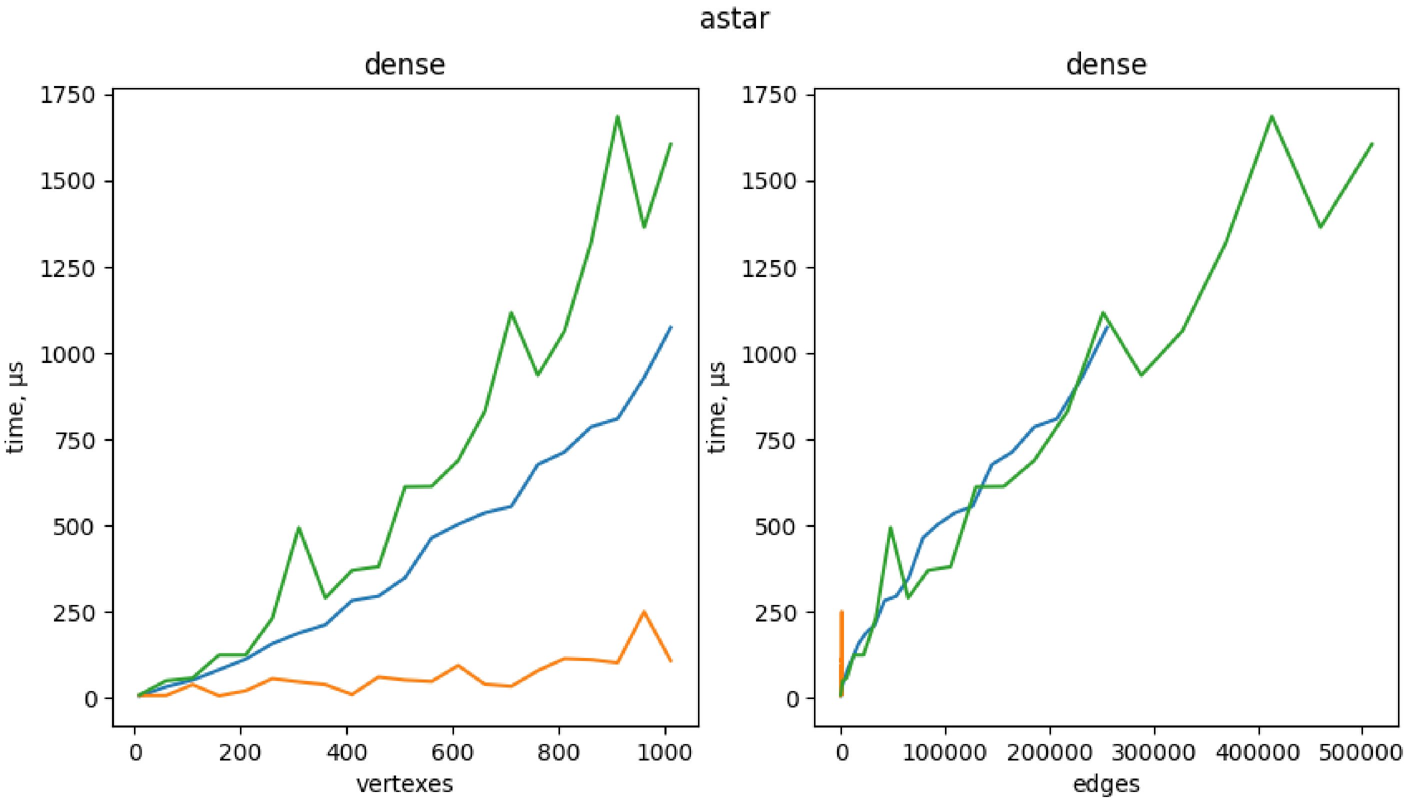
```
for algo in ALGOS:
    # subplot
    fig, axs = plt.subplots(1, 2, figsize=(10, 5))
    fig.suptitle(algo)

    algo_df = pd.DataFrame(columns=['vertexes', 'edges'])
    algo_df.set_index(['vertexes', 'edges'], inplace=True)
    for name, df in data.items():
        algo_df = pd.merge(algo_df, df[algo], left_index=True, right_index=True, how='outer')
    algo_df.columns = [*data.keys()]

    # plot by vertexes
    df_by_vertexes = algo_df.copy()
    df_by_vertexes.index = df_by_vertexes.index.map(lambda x: x[0])
    df_by_vertexes.index.name = 'vertexes'
    df_by_vertexes = df_by_vertexes.groupby('vertexes').mean()
    for col in df_by_vertexes.columns:
        mask = df_by_vertexes[col].isna()
        # plot to subplot
        pl = df_by_vertexes[~mask][col].plot(title=col, ax=axs[0])
        pl.set_xlabel('vertexes')
        pl.set_ylabel('time, µs')

    # plot by edges
    df_by_edges = algo_df.copy()
    df_by_edges.index = df_by_edges.index.map(lambda x: x[1])
    df_by_edges.index.name = 'edges'
    df_by_edges = df_by_edges.groupby('edges').mean()
    for ind, col in enumerate(df_by_edges.columns):
        mask = df_by_edges[col].isna()
        # plot to subplot
        pl = df_by_edges[~mask][col].plot(title=col, ax=axs[1])
        pl.set_xlabel('edges')
        pl.set_ylabel('time, µs')

    plt.show()
```


dijkstras
dense
dense
time, µs
0
1000
2000
3000
4000
5000
6000
0
200
400
600
800
1000
vertexes
0
100000
200000
300000
400000
500000
edges


bellmanford
dense
dense
1e6
1e6
time, µs
0.0
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
vertexes
0
100000
200000
300000
400000
500000
edges


floydwarshall
dense
dense
time, µs
400000
600000
800000


200000
0
0
200
400
600
800
1000
vertexes
200000
0
0
100000
200000
300000
400000
500000
edges


astar
dense
dense
time, µs
1750
1500
1250
1000
750
500
250
0
0
200
400
600
800
1000
vertexes
time, µs
1750
1500
1250
1000
750
500
250
0
0
100000
200000
300000
400000
500000
edges