
Aluno: Matrícula: Valor: 25,0 Nota:

Trabalho Prático I

1. Objetivos.

- Desenvolver a habilidade de leitura e manipulação de dados.
- Reforçar o aprendizado sobre estruturas de dados em grafos.
- Aplicar os conhecimentos em grafos para modelar problemas reais.

2. Descrição.

A Câmara dos Deputados é responsável pela elaboração e votação de leis, além de exercer o controle e fiscalização do Poder Executivo. O Brasil tem vivenciado um momento de grande polarização política, o que frequentemente dificulta a aprovação de pautas importantes para o desenvolvimento do país. Para aprovar um projeto de lei (PL) é necessária aprovação pela maioria dos deputados presentes na câmara. Nesse contexto, é comum os políticos buscarem apoio de seus pares para formar uma maioria e aprovar uma pauta que consideram importante ~~si mesmos~~ para a sociedade.

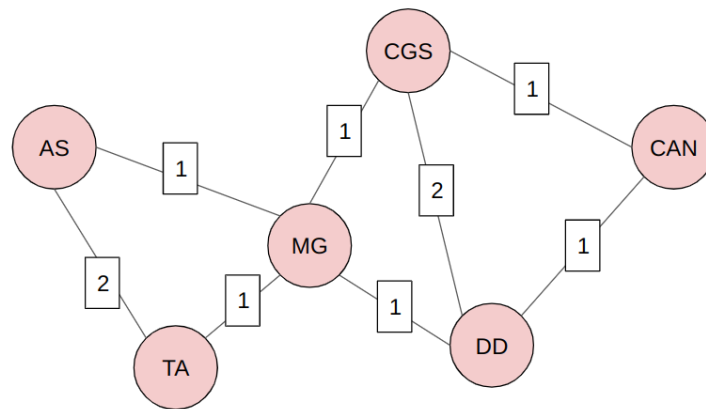
O portal de dados abertos da câmara ¹ disponibiliza uma API (*Application Programming Interface*) contendo diversos dados relacionados ao órgão e seus membros, como deputados, partidos, votações, frentes parlamentares, etc. Além do acesso online, os dados também estão disponíveis para download direto. Com base nesses dados, nesse trabalho iremos construir um grafo que representa as relações de proximidade na Câmara de Deputados.

No trabalho proposto, cada nó representa um deputado d , e cada aresta (d_i, d_j) representa o número vezes que esses deputados votaram em acordo. Caso os deputados nunca tenham votado em acordo em nenhuma votação, não haverá aresta entre eles. Como exemplo, considere o seguinte trecho do arquivo `votacoesVotos-2023.xlsx`:

idVotacao	voto	deputado_id	deputado_nome
1713431-105	Sim	204501	Alencar Santana
1713431-105	Não	220574	Cabo Gilberto Silva
1713431-105	Não	204501	Capitão Alberto Neto
1713431-105	Não	220705	Deltan Dallagnol
1713431-105	Sim	220584	Murilo Galdino
1713431-105	Sim	204534	Tabata Amaral
1713431-120	Sim	204501	Alencar Santana
1713431-120	Não	220705	Deltan Dallagnol
1713431-120	Sim	204534	Tabata Amaral
1713431-125	Sim	204501	Alencar Santana
1713431-125	Não	220574	Cabo Gilberto Silva
1713431-125	Não	220705	Deltan Dallagnol
1713431-125	Não	220584	Murilo Galdino

¹<https://dadosabertos.camara.leg.br/>

Nesse trecho temos 6 deputados e três votações. Naturalmente, nem todos os deputados participaram de todas as votações. A figura a seguir apresenta o grafo correspondente ao arquivo anterior. Note que o nome dos parlamentares foi abreviado para melhor visualização.



Como saída, seu programa deve escrever o grafo em um arquivo de texto de modo análogo ao exemplo abaixo, onde na primeira linha temos o número de nós e de arestas e nas linhas seguintes uma aresta e seu peso separados por espaço:

```

6 8
Alencar_Santana Murilo_Galdino 1
Alencar_Santana Tabata_Amaral 2
Murilo_Galdino Tabata_Amaral 1
Murilo_Galdino Cabo_Gilberto_Silva 1
Murilo_Galdino Deltan_Dallagnol 1
Cabo_Gilberto_Silva Deltan_Dallagnol 2
Cabo_Gilberto_Silva Capitão_Alberto_Neto 1
Deltan_Dallagnol Capitão_Alberto_Neto 1

```

e, em outro arquivo, o número de votações que cada deputado participou, separados por espaço:

```

Alencar_Santana 3
Cabo_Gilberto_Silva 2
Capitão_Alberto_Neto 1
Deltan_Dallagnol 3
Murilo_Galdino 2
Tabata_Amaral 2

```

O arquivo contendo os dados de entrada reais de votações em 2023 pode ser obtidos em <https://dadosabertos.camara.leg.br/swagger/api.html#staticfile>. Pode-se proceder à leitura do arquivo no formato de sua preferência dentre os disponíveis na página (csv, xlsx, xls, json, xml, ods). De acordo com as aulas práticas, o programa deve ser escrito em Python. O uso e adequação do código (classes e funções) disponibilizado em aula é opcional. Como o objetivo é programar as estruturas de dados, e, futuramente, algoritmos para o problema, o uso de bibliotecas para manipulação de grafos é vetado (ex.: NetworkX, igraph, Graph-tool, PyGraphistry, SNAP.py). Já demais bibliotecas são permitidas (ex.: Pandas, numpy, Matplotlib, etc).

No Trabalho II exploraremos esses grafos para tirar conclusões interessantes, como similaridade entre posicionamentos políticos, quais deputados são mais centrais para atingir apoio (baseado em medidas de centralidade), e deputados foram mais assíduos nas votações, dentre outros.

3. Interação com o usuário

A interação com o usuário deve ocorrer no arquivo `main` do seu programa. O mesmo deve solicitar ao usuário o arquivo de entrada e, após a execução, informar o nome dos arquivos de saída escritos. Segue um exemplo de interação com o programa:

```
Informe o arquivo de votações: <votacaoVotos-2023.xlsx>
Processando...

0 grafo foi escrito nos arquivos:
- votacaoVotos-2023-graph.txt
- votacaoVotos-2023-deputados.txt
```

4. Avaliação.

O trabalho deverá ser feito individualmente ou em dupla e enviado via Moodle até as 23:59h do dia 14/07/23. Caso se tenha alguma dúvida com relação à autoria do trabalho o professor poderá solicitar uma apresentação presencial ao aluno (ou dupla).

5. Pontos extra.

A integração de sistemas através do consumo de APIs (*Application Programming Interfaces*) é uma habilidade altamente requisitada no desenvolvimento de sistemas modernos. Serão atribuídos até 5 pontos extra para os trabalhos que consultarem os dados de votações diretamente da API do site dados abertos câmara². O uso de assistentes de IA (como chatGPT) não só é permitido, como é recomendado para auxiliar no desenvolvimento dessa tarefa.

Vocês devem realizar uma requisição `GET /votacoes` para obter os dados de todas as votações. Essa requisição retornará um objeto JSON³ com os dados de todas as pautas de votação registradas no sistema. Posteriormente, para cada votação, através do seu `id`, você deve acessar a rota `GET /votacoes/{id}/votos` para obter dados da votação de cada deputado nessa votação em particular. Com esses dados em mãos, crie o grafo conforme descrito no enunciado, invés de usar o arquivo estático como fonte de dados.

Bom trabalho!

²<https://dadosabertos.camara.leg.br/swagger/api.html#api>

³https://www.alura.com.br/artigos/o-que-e-json?gclid=CjwKCAjwkeqkBhAnEiwA5U-uMwBR6LShpte9ux9yFz3d_COU-dBALJOU7J1oUDGCz0ZheV-eWg0MqxoCuowQAvD_BwE