

Projeto 2
Teoria e Aplicação de Grafos, Turma A, 2025/2
Prof. Díbio

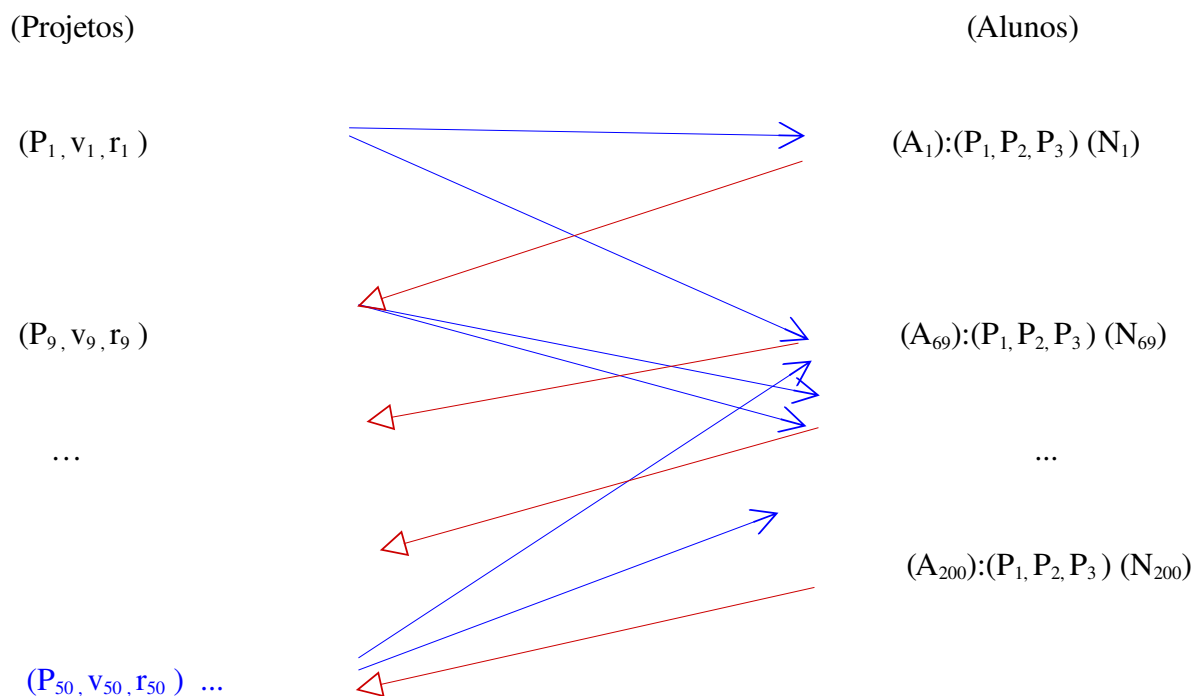
Considere para efeito deste projeto que uma determinada universidade oferece anualmente uma lista de cinquenta (50) projetos financiados e abertos a participação de alunos, com um máximo de 80 vagas. Cada projeto é orientado e gerenciado por professores que estabelecem as quantidades mínima e máxima de alunos que podem ser aceitos em determinado projeto, bem como requisitos de histórico e tempo disponível que os alunos devem possuir para serem aceitos. Esses requisitos de histórico e tempo são pré-avaliados e cada aluno possui uma Nota agregada inteira de [3, 4, 5], sendo 3 indicando suficiente, 4 muito boa, e 5 excelente. Neste ano, duzentos (200) alunos se candidataram aos projetos. O ideal é que o máximo de projetos sejam realizados, mas somente se o máximo de alunos qualificados tenham tido o interesse para tal. Para uma seleção impessoal e competitiva um algoritmo que realize um **emparelhamento estável máximo** deve ser implementado. Este projeto pede a elaboração, implementação e testes com a solução final de emparelhamento máximo estável para os dados fornecidos. Os alunos podem indicar no máximo três (3) preferências em ordem dos projetos. Cada projeto deve obrigatoriamente selecionar pelo menos 1 aluno. Variações do algoritmo Gale-Shapley devem ser usadas, com uma descrição textual no arquivo README do projeto indicando qual variação/lógica foi utilizada/proposta.

O programa deve implementar uma interface com visualização colorida do grafo bipartido e das soluções propostas de emparelhamento, ao longo da evolução com dez (10) iterações de emparelhamento em laço, organizando as saídas até a alocação final. Escolha cores diferentes para indicar arestas: 1) proposta ativa; 2) emparelhamento temporário; 3) rejeição. Um índice de preferência por projeto deve ser calculado, bem como visualizada uma matriz final dos emparelhamentos com as ordens de escolhas de emparelhamentos finais dos alunos.

A matriz de emparelhamento deve incluir **não apenas a ordem de escolha do aluno**, mas também o **rank do aluno na lista de preferência do projeto**. Isto visualiza o "ganho" ou "perda" para cada lado. exemplos

Aluno	Projeto Emparelhado	Rank do Aluno (Lista de Preferência do Projeto)	Rank do Projeto (Lista de Preferência do Aluno)
105	Projeto C	3° (Top 3 de 15 alunos na lista C)	1° (Primeira escolha do Aluno 105)
042	Projeto A	1° (Top 1 de 8 alunos na lista A)	3° (Terceira escolha do Aluno 042)

As soluções dadas em (Abraham, Irving & Manlove, 2007) são úteis e qualquer uma pode ser implementada com variações pertinentes. Um arquivo entradaProj2.25TAG.txt com as indicações de código do projeto, número de vagas, requisito mínimo das vagas, bem como dos alunos com suas preferências de projetos na ordem e suas notas individuais, é fornecido como entrada. Uma versão pública do artigo de (Abraham, Irving & Manlove, 2007) é fornecida para leitura.



Referência:

Abraham, D.J. and Irving, R.W. and Manlove, D.F. (2007). Two algorithms for the student-project allocation problem. *Journal of Discrete Algorithms* 5(1):pp. 73-90.

Após a complementação do código, escrevê-lo para entrega em um notebook Jupyter, ou google/colab, com comentários, instruções em todas as etapas e nomes e matrículas de todos os integrantes (dupla ou tripla). Um mini-relatório de até 5 páginas deverá ser entregue com a explicação do projeto no aprender3.unb.br, e uma demonstração deve ser feita em sala de aula a pedido do professor. Os prazos serão indicados no <http://aprender3.unb.br> do curso.