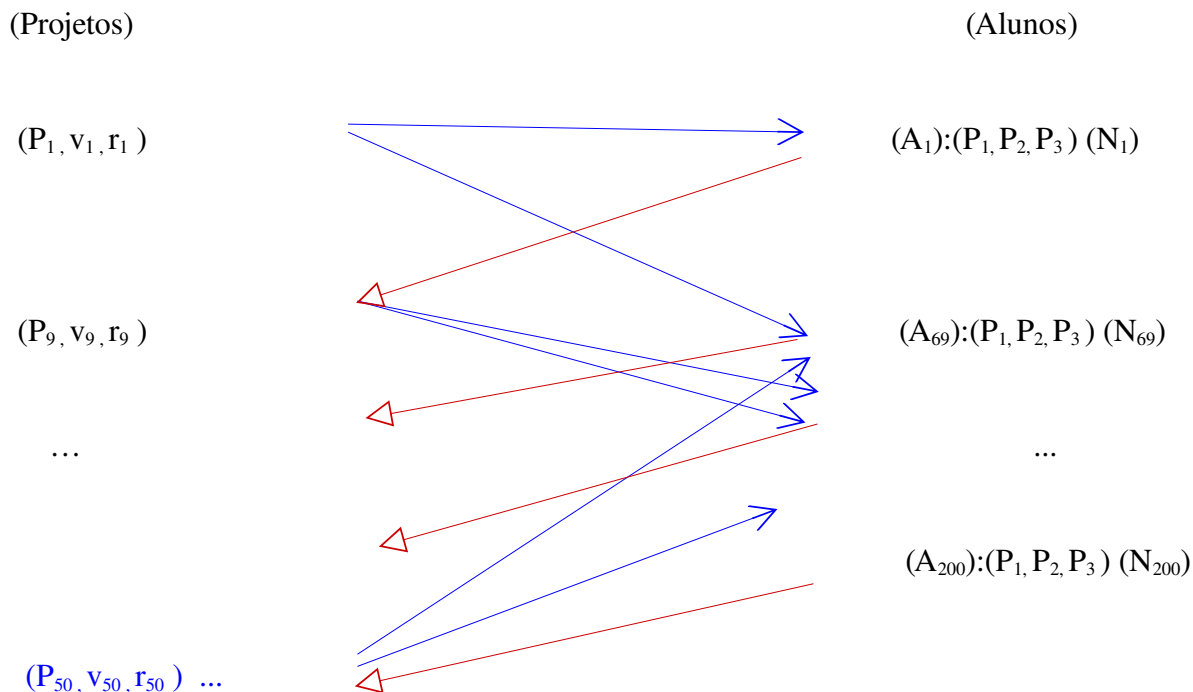


Projeto 2  
Teoria e Aplicação de Grafos, Turma A, 2023/2  
Prof. Díbio

Considere para efeito deste projeto que uma determinada universidade oferece anualmente uma lista de cinquenta (50) projetos financiados e abertos a participação de alunos. 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. 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 tentar, e mostrar todas na tela, dez (10) iterações de emparelhamento em laço, organizando as saídas até a alocação final. Os pares Projetos x Alunos finais devem ser indicados, bem como os números finais. As soluções dadas em (Abraham, Irving & Manlove, 2007) são úteis e qualquer uma pode ser implementada com variações pertinentes. Um arquivo entradaProj2TAG.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.



Para efeito de avaliação, a pontuação será a seguinte:

- a) Três (3) pontos para código modular, comentado, e individual, que lê o arquivo de entrada, monta a solução como emparelhamento de grafo bipartido, e compila e gera executável sem erros de compilação;
- b) Três (3) pontos para solução com emparelhamento estável;
- c) Quatro (4) pontos para solução com emparelhamento **máximo e estável** (iterações com trocas e aumentos).

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.