**ALGORITMOS I**

**Trabalho Prático 2**

Gustavo Amaral Bernardino – 2021432674

Departamento de Ciência da Computação

Universidade Federal de Minas Gerais

(DCC / UFMG)

Belo Horizonte – MG – Brasil

gustavo.amaral@dcc.ufmg.br

# INTRODUÇÃO

A situação-problema a ser solucionada pelo presente TP tem como pano de fundo uma empresa fictícia que conecta candidatos e vagas de emprego. A empresa está tendo dificuldades em direcionar as vagas aos candidatos de forma que cada vaga seja apresentada a somente um candidato e que cada candidato seja oferecido somente uma vaga. Neste sentido, deve ser criado um algoritmo que indique quantos pares únicos de candidato-vaga podem ser formados diante de determinada relação de usuários e de vagas.

Além disso, o TP ainda propõe que sejam desenvolvidas duas soluções para o problema, sendo:

1. Uma gulosa, que pode ser subótima, mas que precisa encontrar a solução de forma rápida;
2. Uma exata, que precisa retornar a melhor solução possível ao problema,

# MODELAGEM

## Estruturas de Dados Utilizadas

A estrutura de dados escolhida para representar o grafo foi uma matriz de adjacência, cujas linhas representam os vértices de usuários e as colunas os vértices de vagas, enquanto o valor de cada posição indica se aquele usuário é apto para aquela vaga, sendo 1 = sim e 0 = não. A opção por tal estrutura se deu principalmente pela maior facilidade de implementação quando comparado com outros tipos de implementação, apesar da tendência de maior custo de espaço.

A outra estrutura de dados utilizada para a resolução do problema foram os mapas, por meio da biblioteca <map>. Essa estrutura foi utilizada para auxiliar na transformação das strings dos nomes dos usuários e das vagas em índices que poderiam ser utilizados na matriz de adjacência. Assim, o programa conta com dois mapas, um para cada conjunto, no qual são armazenados os dicionários dos dados do grafo.

## Implementação dos Algoritmos Principais

Como mencionado anteriormente, o TP exigiu a construção de dois algoritmos para a solução do problema principal, sendo um Guloso e outro Exato.

O Algoritmo Guloso funciona iniciando a análise pelo lado das vagas. Nesse sentido, ele seleciona a primeira vaga e identifica se há algum candidato apto a ela. Caso exista, esse candidato é imediatamente alocado na vaga e o algoritmo registra que o candidato passou a estar indisponível, e a avaliação passa para a vaga seguinte.

O processo anterior é repetido para esta vaga, porém agora é possível que o primeiro candidato apto para a vaga seja aquele que foi alocado na primeira. Assim, o algoritmo conta também com a avaliação da disponibilidade do candidato, para além da sua aptidão para a vaga. Logo, um candidato somente é alocado para determinada vaga se ele ainda não tiver sido alocado para nenhuma outra e, depois de alocado, não há mais alteração.

Enquanto isso, o Algoritmo Exato funciona de maneira mais sofisticada, uma vez que adiciona um elemento de recursividade que possibilita que sejam realizadas trocas de alocação de vaga-candidato após a alocação inicial. Para isso, o algoritmo começa passando por cada candidato e verifica para quais vagas ele é apto.

Caso a vaga esteja disponível, ele é alocado na vaga. No entanto, se a vaga já está ocupada por outro candidato, o algoritmo realiza uma nova busca para esse outro candidato, verificando se existe alguma vaga para a qual ele seja apto e que ele ainda não explorou. Isso é feito recursivamente, de forma que caso a nova vaga disponível já esteja ocupada por um terceiro candidato, então o algoritmo é chamado novamente.

Quando todas as chamadas recursivas são retornadas, o resultado é um booleano que indica se é possível rearranjar os candidatos envolvidos de forma a manter todos alocados a uma vaga. Somente em caso positivo – isto é, se todos puderem ser movimentados – é feita a alocação do primeiro candidato e de todos os demais.

## Comparação dos Algoritmos

Como esperado, o resultado do Algoritmo Guloso foi subótimo quando comparado ao Exato, apesar de ter obtido a resposta ideal em alguns poucos testes. Isso ocorres porque, como citado anteriormente, o Guloso não permite que uma alocação usuário-vaga seja desfeita momentaneamente em prol de outra possivelmente mais benéfica.

Partindo para a comparação em termos de complexidade, primeiramente é preciso definir a notação utilizada:

Isto posto, tem-se que o Algoritmo Guloso realiza a inicialização de dois vetores auxiliares, um de tamanho e outro de tamanho . Em seguida, realiza um loop por toda a matriz de adjacência de tamanho e, por fim, efetua um novo laço sobre o vetor de tamanho . Com isso, a complexidade total é de:

Enquanto isso, o Algoritmo Exato inicializa um vetor auxiliar de tamanho e inicializa um vetor auxiliar de tamanho para cada um dos candidatos. Para cada um dos m candidatos há também uma chamada da função recursiva. No pior caso, todos os usuários são aptos para todas as vagas, então a função recursiva sempre é chamada. Nessa situação,

Por fim, também é efetuado um último laço sobre o vetor de tamanho . Diante disso, a complexidade total é de:

Com isso, é possível observar que o algoritmo exato possui uma complexidade consideravelmente maior que o guloso.