

# ACH2024 ALGORITMOS E ESTRUTURAS DE DADOS II

## Semestre 2025-1 - Exercício prático – Grafos – t.02

**Contexto:** O objetivo do EP é encontrar vértices equidistantes a dois vértices dados considerando o caminho mais curto possível a partir deles.

**Descrição do EP:** A partir do *modelo.txt* disponibilizado com esta atividade, implementar a seguinte função *equidistantes* sobre uma estrutura de dados do tipo grafo não-dirigido:

`NO *equidistantes(int V, int A, int ij[], int v1, int v2);` // corrigido: antes dizia `int* v1` e `int *v2` (altere na assinatura de *modelo.txt* também, ou use a versão que está no e-disciplinas, que já está corrigida.)

### Grafo de entrada:

- Os parâmetros  $V$ ,  $A$  e o vetor  $ij[]$  definem um grafo de  $V$  vértices numeradas de 1 a  $V$  e  $A$  arestas. O vetor  $ij[]$  contém exatamente  $2*A$  elementos, e cada par representa uma aresta  $i,j$ .
- Os vértices são numerados de 1 a  $V$  (portanto, crie um vetor de  $V+1$  posições e despreze a posição zero), mas note que o vetor  $ij[]$  é apenas uma lista de  $A$  pares de vértices, e você deve considerar todos os pares desde a posição zero do vetor.

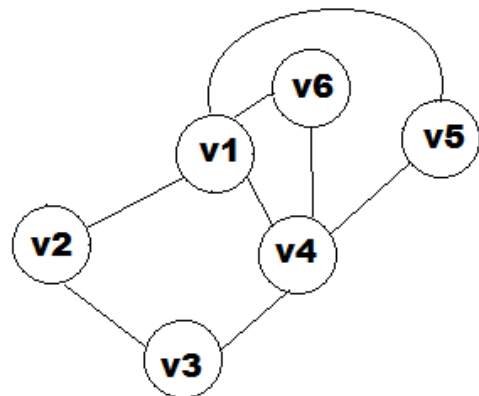
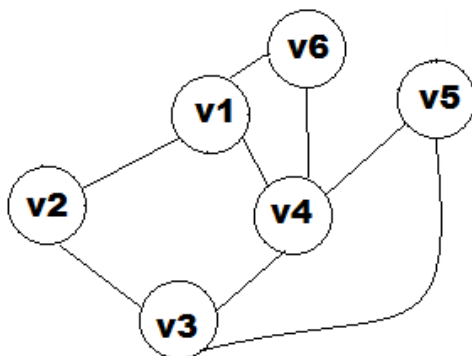
### Função a ser implementada:

Seja um grafo não dirigido a ser construído a partir dos parâmetros  $V$ ,  $A$  e  $ij[]$  descritos acima, e dois vértices  $v1$  e  $v2$ , encontre todos os vértices equidistantes a  $v1$  e  $v2$  considerando o caminho mais curto possível de  $v1$  ou  $v2$  até cada um dos vértices de resposta.

A função deve retornar uma lista ligada com a resposta em qualquer ordem, e sem incluir  $v1$  ou  $v2$ . Havendo mais de um vértice que atenda os critérios, todos devem constar na lista sem repetição. Se não houver nenhum vértice que atenda aos critérios, a lista retornada deve ser vazia.

### Exemplos:

No grafo da esquerda a seguir, dados os vértices de início  $v1$  e  $v2$ , a resposta seria  $\{v5\}$ , que é o único vértice que está exatamente à mesma distância tanto de  $v1$  como de  $v2$ . Já no grafo da direita, a resposta é uma lista vazia  $\{NULL\}$ , pois não existe nenhum vértice que esteja a exatamente a mesma distância tanto de  $v1$  como de  $v2$  simultaneamente.



Para manipulação desta estrutura pelo EP, sugere-se assim que seja primeiramente criado um grafo (em listas de adjacências ou em matriz) a partir deste conjunto de parâmetros. SE CRIAR UM GRAFO EM VETOR, NÃO

UTILIZE A POSIÇÃO ZERO OU SEU EP SERÁ INVALIDADO. O GRAFO NÃO PODE TER VÉRTICE ZERO.

Para ser considerada válida, a lista fornecida como resposta deve obrigatoriamente atender aos seguintes requisitos.

- A lista de resposta deve seguir o formato especificado no projeto, com um inteiro (adj) em cada nó.
- Não use nó cabeça, sentinela, circularidade ou encadeamento duplo. É uma lista ligada simples do tipo NO fornecido.
- Se não existir nenhuma resposta possível, a lista fica vazia (NULL).
- Se existir uma resposta não nula, ela deve ser composta apenas de vértices válidos sem repetição, em qualquer ordem.

Tenha em mente que **tudo que for necessário** para executar a função solicitada deve obrigatoriamente estar no arquivo entregue, ou seu EP estará incompleto.

Não exiba nenhuma mensagem na tela, nem solicite que o usuário pressione nenhuma tecla etc. Apenas implemente a função solicitada.

A função *main()* serve apenas para seus testes particulares, e não deve ser entregue.

Seu programa será corrigido de forma *automática*, e por isso você não pode alterar as assinaturas da função solicitada, nem os tipos de dados ou especificações (*typedef*) do modelo fornecido.

O EP pode ser desenvolvido individualmente ou em duplas, desde que estas sejam cadastradas até **27 de março**. Duplas formadas tardiamente não serão consideradas. Alunos que queiram trabalhar individualmente também devem se cadastrar. Por favor acesse o link abaixo e cadastre seus dados usando sua conta USP:

<https://docs.google.com/spreadsheets/d/1LEgFCYPV232H2Cggp7Datvx55G4DCpaVqwtH29Syll/edit?usp=sharing>

Não tente emprestar sua implementação para outros colegas, nem copiar deles, pois isso invalida o trabalho de todos os envolvidos.

O programa deve ser compilável no Codeblocks 13.12 sob Windows 10 ou superior. Será aplicado um desconto de 30% na nota do EP caso ele não seja imediatamente compilável nesta configuração.

#### O que/como entregar:

- A entrega será via upload no sistema e-disciplinas por **apenas um** integrante de cada dupla.
- Entregue apenas o código da função principal e das funções auxiliares que ela invoca em um arquivo trabalho.cpp - **favor não compactar**.
- Preencha as funções *nroUSP1()* e *nroUSP2()* do modelo disponível no sistema para que você seja identificado. Se o EP for individual, mantenha o valor do segundo nro. como zeros.
- Na primeira linha do código, escreva um comentário `"/"` com os nomes dos integrantes.

#### Prazos e penalidades:

O EP deve ser depositado no prazo definido na atividade cadastrada no sistema. Não serão aceitos EPs entregues depois do prazo, independentemente do motivo. Entregas no último dia são assim por conta e risco do aluno, e nenhum tipo de imprevisto de última hora (e.g., problemas de saúde, indisponibilidade de rede etc.) pode ser usado como justificativa para o atraso. O EP é uma atividade para ser desenvolvida ao longo de várias semanas, não nos últimos dias antes da entrega. É responsabilidade do aluno que fez o *upload* do arquivo verificar se o mesmo foi corretamente recebido pelo sistema. Atrasos/falhas na submissão invalidam o trabalho realizado.

Após o *upload*, verifique se você consegue abrir o arquivo que está no sistema, e certifique-se de que é a versão correta do programa.

**Critérios de avaliação:**

A função será testada com uma série de chamadas repetidas e consecutivas, com diversos tipos de entradas. É assim importante assegurar que o seu programa funciona desta forma (por exemplo, chamando-o dentro de um laço *for*), e não apenas para um teste individual. Um teste é considerado correto se a lista fornecida representar uma resposta válida, ou incorreto em caso contrário. Erros de alocação de memória ou compilação invalidam o teste, assim como a ausência de funções auxiliares necessárias para a execução do programa.

Este EP deve ser desenvolvido obrigatoriamente por *todos* os alunos de AED2. Sua nota é parte integrante da 1ª. avaliação e *não* é passível de substituição.