ORDENAÇÃO TOPOLÓGICA EM GRAFOS

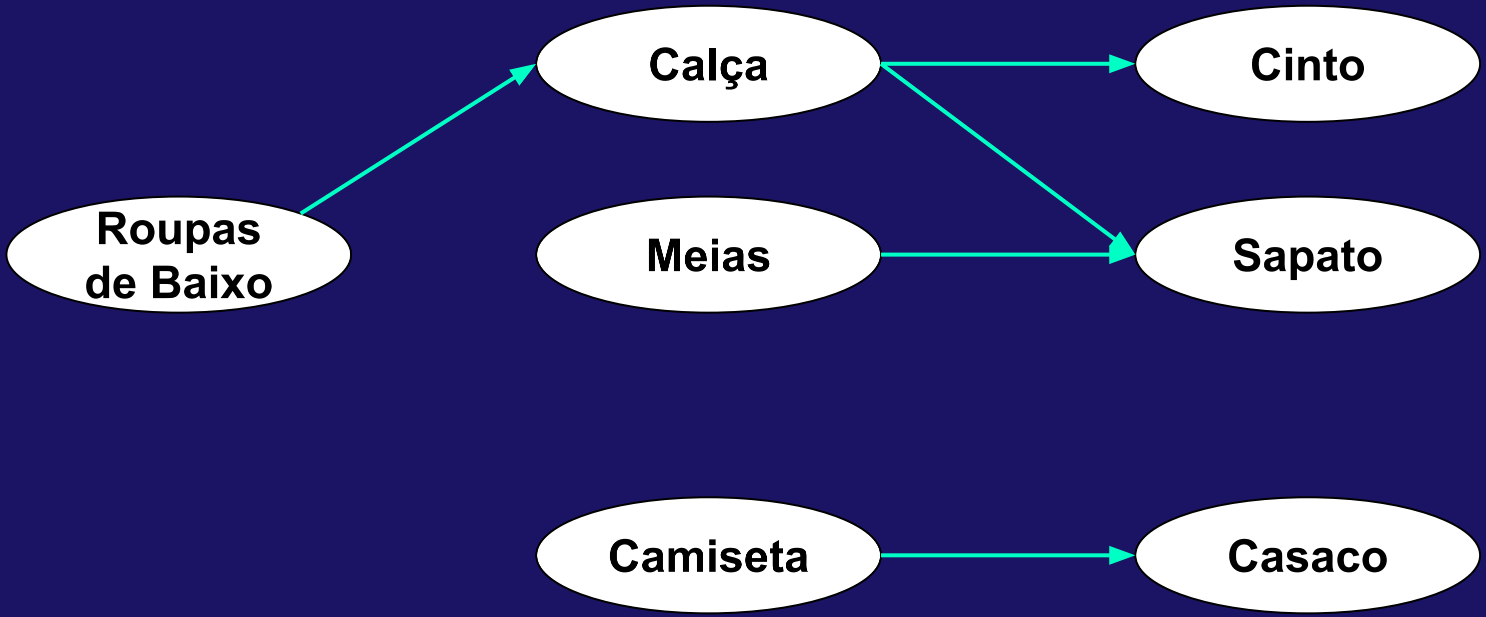
ALUNO: THIAGO THEIRY DE OLIVEIRA

DEFINIÇÃO DE UMA ORDENAÇÃO TOPOLÓGICA

Uma ordenação topológica de um grafo acíclico direcionado (GAD), é uma ordenação linear de seus vértices, na qual cada vértice aparece antes de seus descendentes. Em outras palavras, é uma ordenação linear de vértices na qual cada vértice precede os vértices que formam seu fecho transitivo direto. Cada GAD possui uma ou mais ordenações topológicas. Caso um grafo possua ciclos ou seja não direcionado, não será possível estabelecer uma relação de precedência entre os vértices, e portanto, é impossível estabelecer uma ordenação topológica

Entendendo melhor

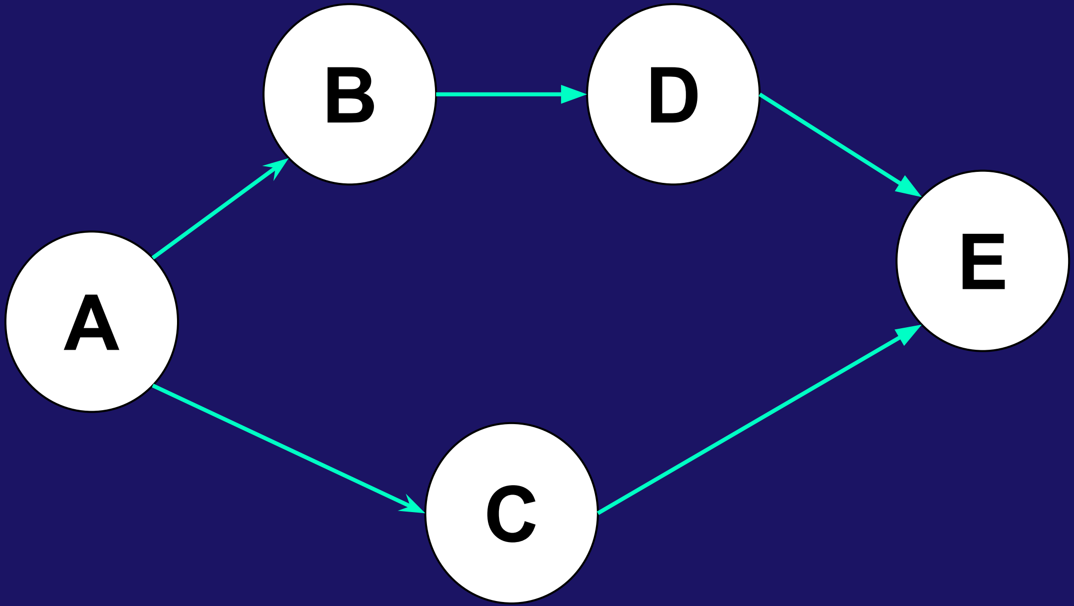
EXEMPLO 1



Entendendo melhor

EXEMPLO 2

- B depende de A
- C depende de A
- D depende de B
- E depende de C
- E depende de D



Ordenação possível : A, B, C, D, E ou A, C, B, D, E

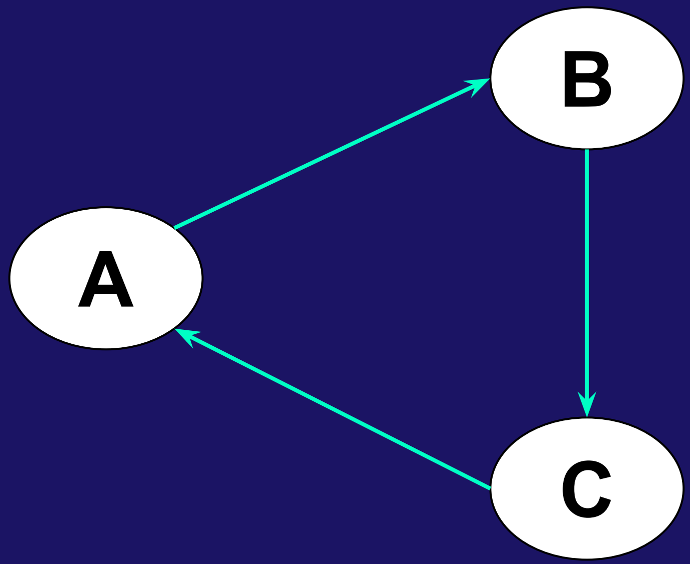
PORQUE A ORDENAÇÃO TOPOLÓGICA TEM QUE SER ACÍCLICA

Porque não será possível estabelecer uma relação de precedência entre os vértices, e portanto, é impossível estabelecer uma ordenação topológica

Entendendo melhor

EXEMPLO 3

- B depende de A
- C depende de B
- A depende de C



Ordenação possível : Não Existe

Algoritmos possíveis para realizar uma implementação de ordenação topológica

Algoritmo de kahn

Busca em profundidade – DFS

</>

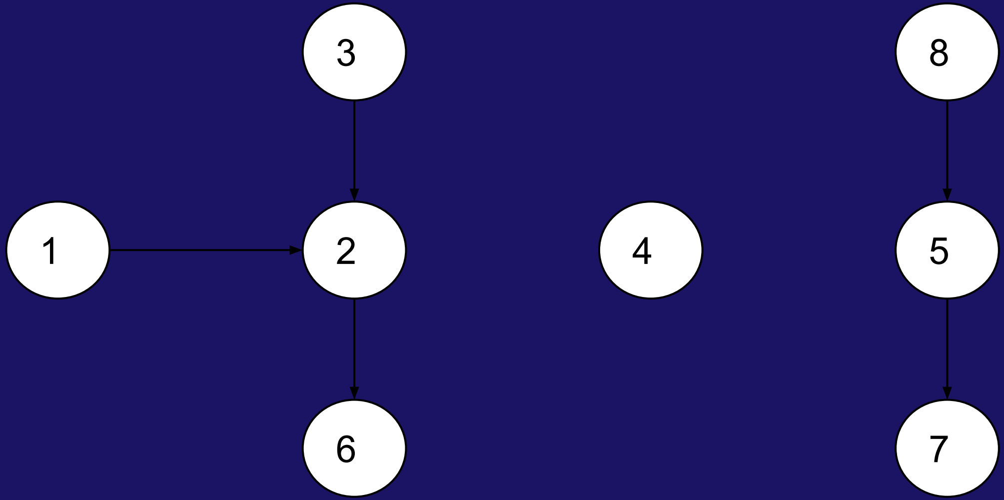
“O algoritmo de Kahn data de 1962 e possui como princípio determinar a cada instante os vértices que não possuam arcos de entrada e inserir na solução. A cada vértice inserido na solução, todos seus arcos correspondentes são removidos do grafo. Também detecta a existência de ciclos no grafo”

ALGORITMO DE KANH

```
L ← Lista vazia que irá conter os elementos ordenados
S ← Conjunto de todos os nós sem arestas de entrada
enquanto S é não-vazio faça
    remova um nó n de S
    insira n em L
    para cada nó m com uma aresta A de n até m faça
        remova a aresta A do grafo
        se m não tem mais arestas de entrada então
            insira  m em S
se o grafo tem arestas então
     escrever mensagem de erro (grafo tem pelo menos um ciclo)
senão
    escrever mensagem  (ordenação topológica proposta: L)
```

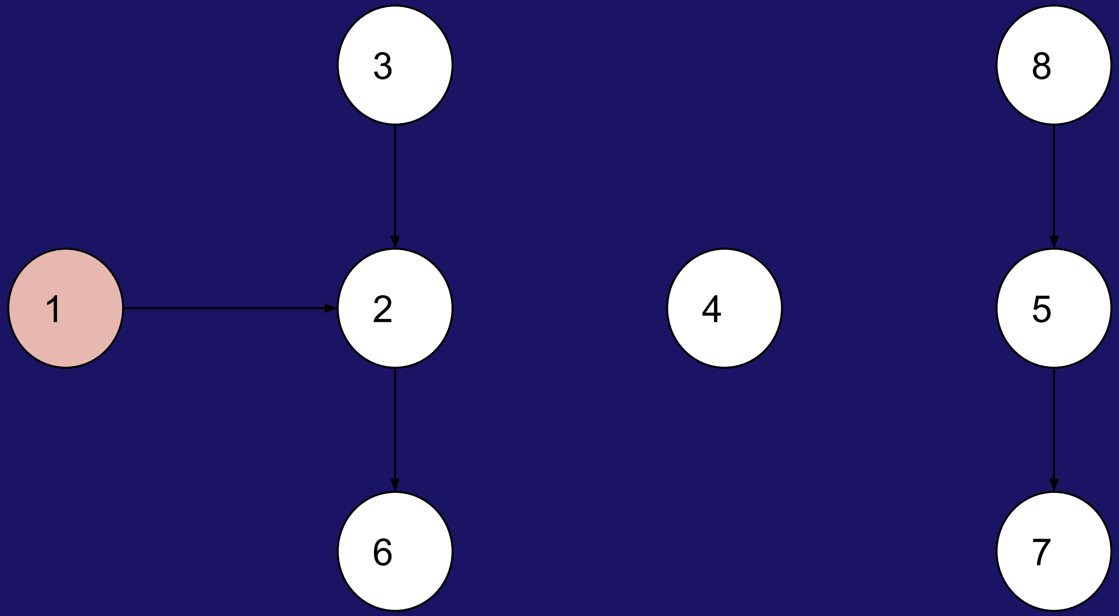
Entendendo melhor

EXEMPLO 4



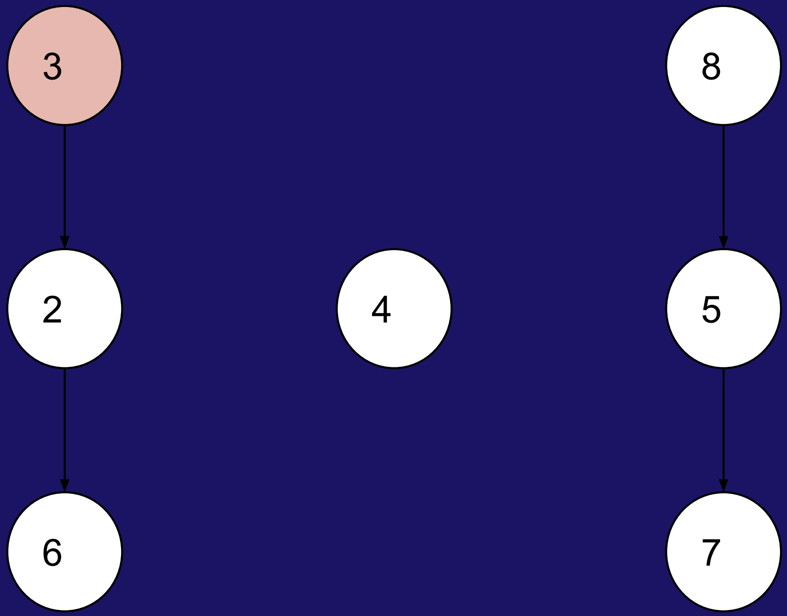
Ordem:

Vértice	Grau
1	0
2	2
3	0
4	0
5	1
6	1
7	1
8	0



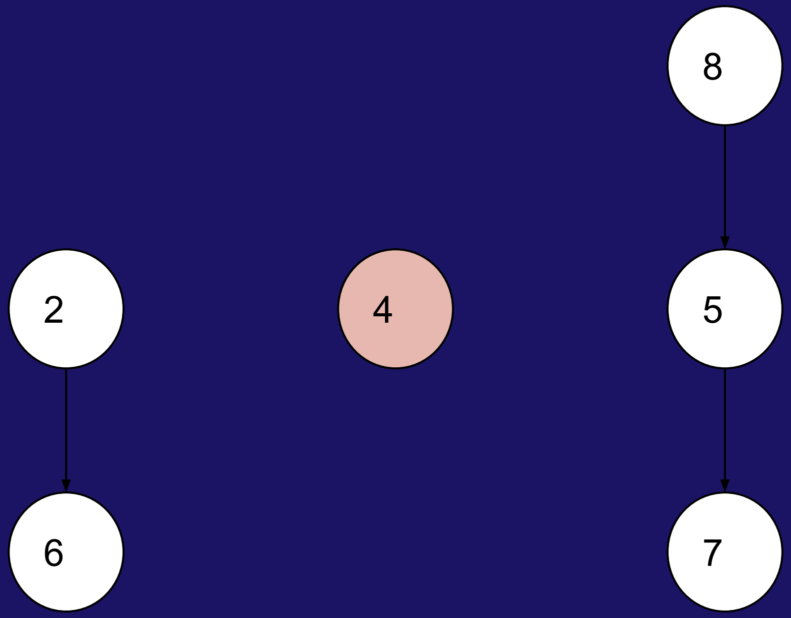
Vértice	Grau
1	-
2	1
3	0
4	0
5	1
6	1
7	1
8	0

Ordem: 1



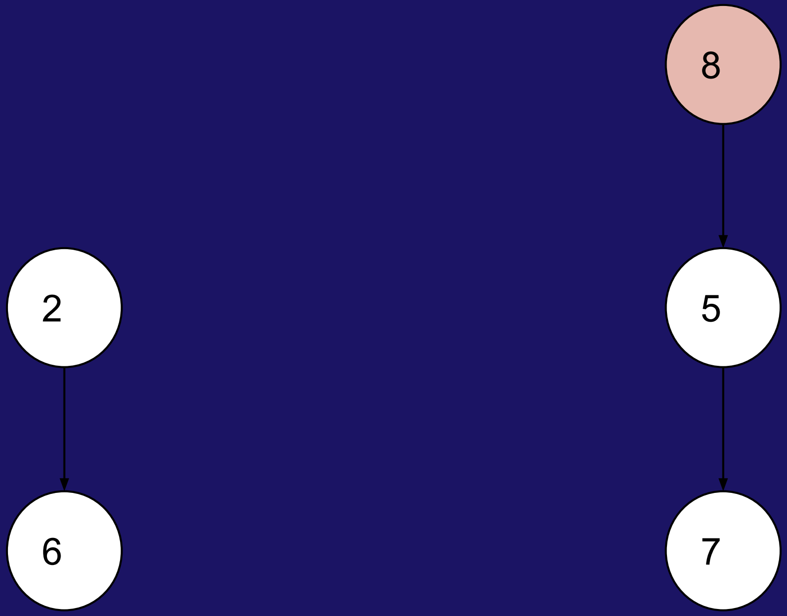
Vértice	Grau
1	-
2	0
3	-
4	0
5	1
6	1
7	1
8	0

Ordem: 1 3



Vértice	Grau
1	-
2	0
3	-
4	-
5	1
6	1
7	1
8	0

Ordem: 1 3 4



Vértice	Grau
1	-
2	0
3	-
4	-
5	0
6	1
7	1
8	-

Ordem: 1 3 4 8

Vértice	Grau
1	-
2	-
3	-
4	-
5	0
6	0
7	1
8	-

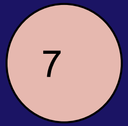
Ordem: 1 3 4 8 2

Vértice	Grau
1	-
2	-
3	-
4	-
5	-
6	0
7	0
8	-

Ordem: 1 3 4 8 2 5

Vértice	Grau
1	-
2	-
3	-
4	-
5	-
6	-
7	0
8	-

Ordem: 1 3 4 8 2 5 6

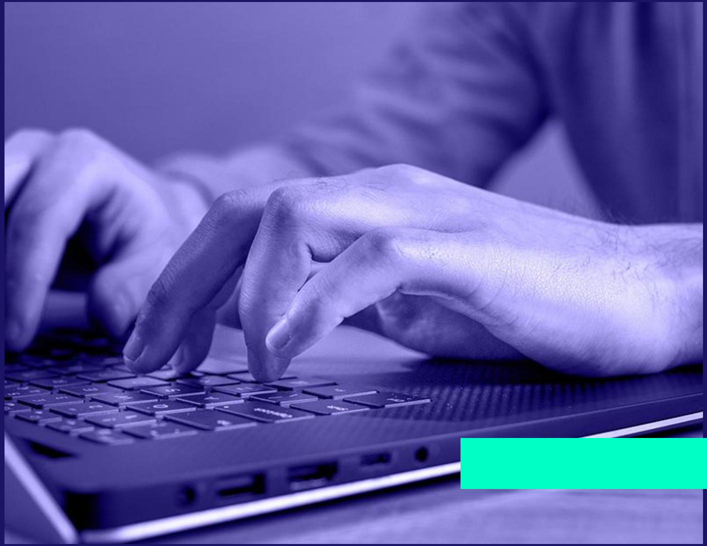


Vértice	Grau
1	-
2	-
3	-
4	-
5	-
6	-
7	-
8	-

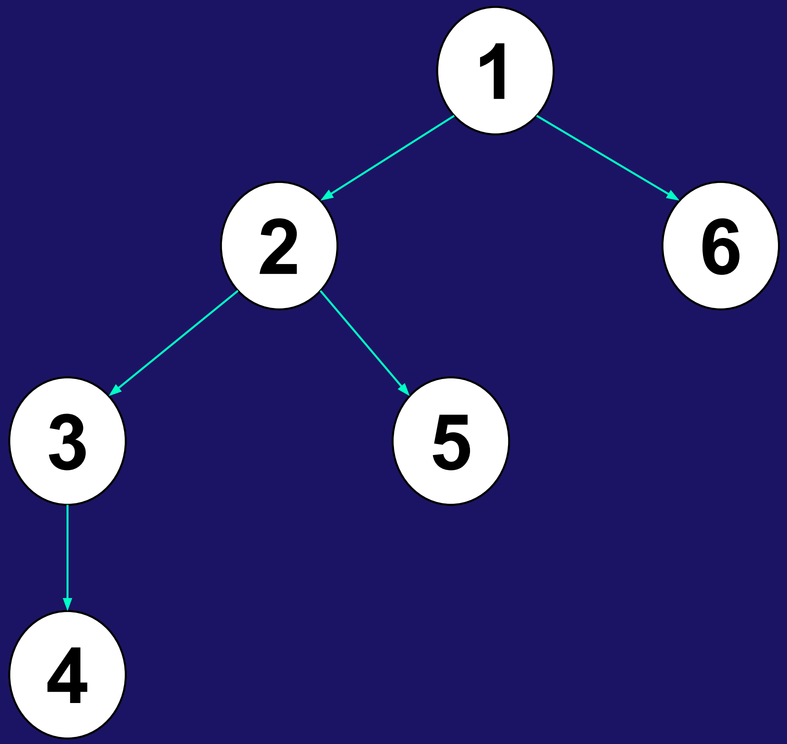
Ordem: 1 3 4 8 2 5 6 7

Relembrando

DFS - Depth-First Search: Explora-se profundamente cada vértice do grafo sempre procurando avançar na busca, sem olhar para os nós vizinhos de mesmo nível. Quando necessário, faz-se backtracking e volta-se aos vizinhos abandonados antes



Funcionamento da DFS



ALGORITMO DA DFS

```
L ← Lista vazia que irá conter os elementos ordenados
S ← Conjunto de todos os nós sem arestas de entrada

função visita(no n)
    se n não foi visitado ainda então
        marque n como visitado
        para cada no m com uma aresta de n para m faça
            visite(m)
        adicione n em L

para cada nó n em S faça
    visite(n)
```

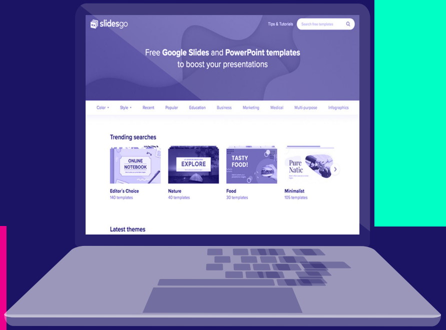
ALGORITMO DA DFS - Detecta Ciclos

1. L: Lista que conterá os elementos da ordenação topológica;
2. Um vértice pode ser não marcado, temporariamente marcado ou definitivamente marcado;
3. Inicialmente, todos os vértices são não marcados, ao serem atingidos pela primeira vez, os vértices são temporariamente marcados;
4. Após terem todas as suas dependências examinadas, os vértices são definitivamente marcados;
5. Caso um vértice temporariamente marcado seja examinado novamente, o grafo possui pelo menos um ciclo

ALGORITMO DA DFS – Detecta Ciclos

```
Entrada: Grafo G = (V, A)
L ← ∅;
Enquanto existir vértice não marcado e sem arcos de
entrada faça
        selecione um vértice v não marcado;
         visite(G, v, L);
fim
função visite(G,v,L)
        se v é temporariamente marcado então retorna Erro;
                se v é não marcado então
                marque temporariamente v;
                para cada arco (vw) faça
                        visite(G, w, L);
                fim
        marque definitivamente v;
        adicione v ao final de L;
 fim
```

Qual é o melhor?

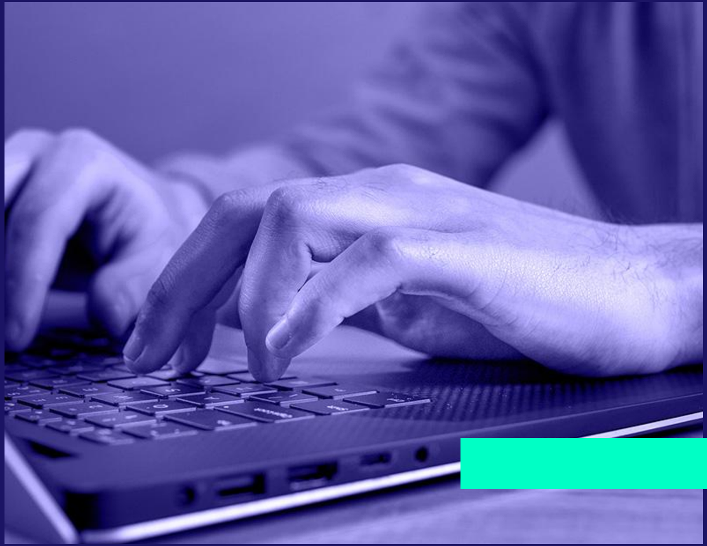


Algoritmo de kahn

Busca em profundidade – DFS

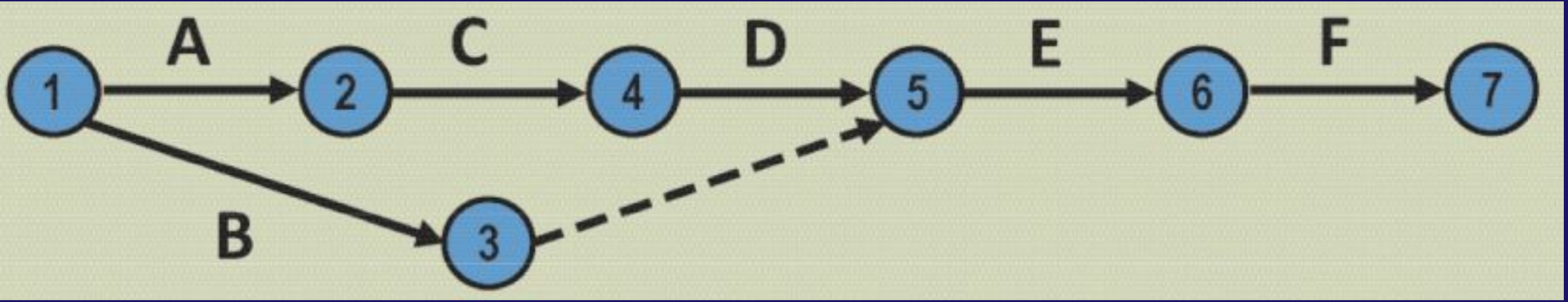
Motivação

Dado um conjunto de N tarefas (dependentes entre si), em que ordem podemos executar estas tarefas.



Exemplo Pratico

Atividade	Discrição	Duração	Anterior	Posterior
A	Comprar tábuas	1 dia	–	3
B	Comprar parafusos	1 dia	–	5
C	Cortar as tábuas	2 dias	1	4
D	Pintar as tábuas	1 dia	2	5
E	Montar as tábuas com parafusos	1 dia	4,2	6
F	Transportar a estante	1 dia	5	–



OBRIGADO!

VOCÊS TEM ALGUMA PERGUNTA?