

# BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

## TEORIA DOS GRAFOS

### Escalonamento de Tarefas<sup>1</sup>

O senhor Pedro e seus irmãos gerenciam uma marcenaria, nas montanhas da Serra do Mar, que fabrica cadeiras de balanço com assentos de almofada. O processo de fabricação pode ser dividido em algumas tarefas, algumas das quais têm como pré-requisito outras tarefas. A tabela a seguir mostra as tarefas envolvidas na fabricação das cadeiras de balanço, seus pré-requisitos e o número de horas necessário para realizar cada uma :

Tarefa	Tarefas pré-requeridas	Horas para realização
1. Seleção da madeira	Nenhuma	3,0
2. Entalhamento dos arcos	1	4,0
3. Entalhamento do assento	1	6,0
4. Entalhamento do encosto	1	7,0
5. Entalhamento dos braços	1	3,0
6. Seleção do tecido	Nenhuma	1,0
7. Costura da almofada	6	2,0
8. Montagem do assento e do encosto	3, 4	2,0
9. Fixação dos braços	5, 8	2,0
10. Fixação dos arcos	2, 8	3,0
11. Aplicação do verniz	9, 10	5,0
12. Colocação da almofada	7, 11	0,5

Um diagrama PERT (*Performance Evaluation and Review Technique*) é um gráfico usado para agendar tarefas, originalmente desenvolvido para nortear a construção de submarinos da marinha, mas útil para o gerenciamento de qualquer projeto complexo que possa ser dividido em subtarefas. O diagrama PERT para a fabricação de cadeiras de balanço é mostrado na figura 1., com os nomes das tarefas representados por seus números e com setas apontando da(s) tarefa(s) pré-requisito(s) para a(s) tarefa(s) seguinte(s).

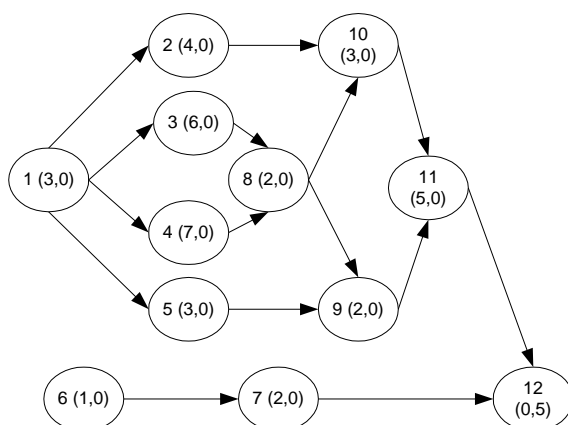


Figura 1 : Diagrama PERT para fabricação de cadeiras de balanço.

Um projeto representado por um diagrama PERT precisa começar pelas tarefas representadas pelos vértices mais à esquerda do diagrama PERT e acabar nas tarefas representadas pelos vértices mais à direita do diagrama. Um limite superior para o tempo necessário para a realização do projeto pode ser obtido pela adição dos tempos de realização de cada tarefa, mas isto não leva em conta o fato de que duas tarefas podem ser realizadas paralelamente, tais como as tarefas 2 e 5. Para obter o tempo mínimo

<sup>1</sup> **Importante:** A especificação desse trabalho pode sofrer modificações de acordo com discussões que tivermos em sala de aula.

necessário para realizar o projeto, podemos caminhar no diagrama, da esquerda para direita, computando o tempo mínimo para a realização do trabalho desde o início do projeto até o término da tarefa representada pelo vértice em questão. Se um vértice  $x$  tem diversos vértices como pré-requisitos, todos esses pré-requisitos devem estar prontos antes que comecemos a tarefa  $x$ ; portanto devemos adicionar ao tempo necessário para a tarefa  $x$  o maior tempo de realização dentre os vértices que são seus pré-requisitos.

O diagrama PERT é um caso particular de aplicação de **ordenação topológica**. Sua importância se deve ao fato de ter um uso potencial todas as vezes que o problema abordado envolve uma ordem parcial, isto é, no qual uma ordenação é efetuada somente sobre alguns pares de elementos e não sobre todo o seu conjunto.

Podemos definir uma **ordenação parcial** no conjunto de tarefas por

$x \leq y \leftrightarrow$  tarefa  $x$  iguala tarefa  $y$  ou tarefa  $x$  é um pré-requisito para tarefa  $y$

Além disso,

$x < y \leftrightarrow$  tarefa  $x$  é um pré-requisito para tarefa  $y$

Dada uma ordenação parcial em um conjunto finito de tarefas, sempre há uma ordenação total que é uma extensão da ordenação parcial. O processo de **ordenação topológica** encontra uma ordenação total deste tipo a partir de uma dada ordenação parcial. Isto é, na verdade, um processo de ordenação na medida em que os elementos terminam totalmente ordenados; mas, uma vez que os elementos precisam estar antes parcialmente ordenados, este é um método de ordenação muito especializado.

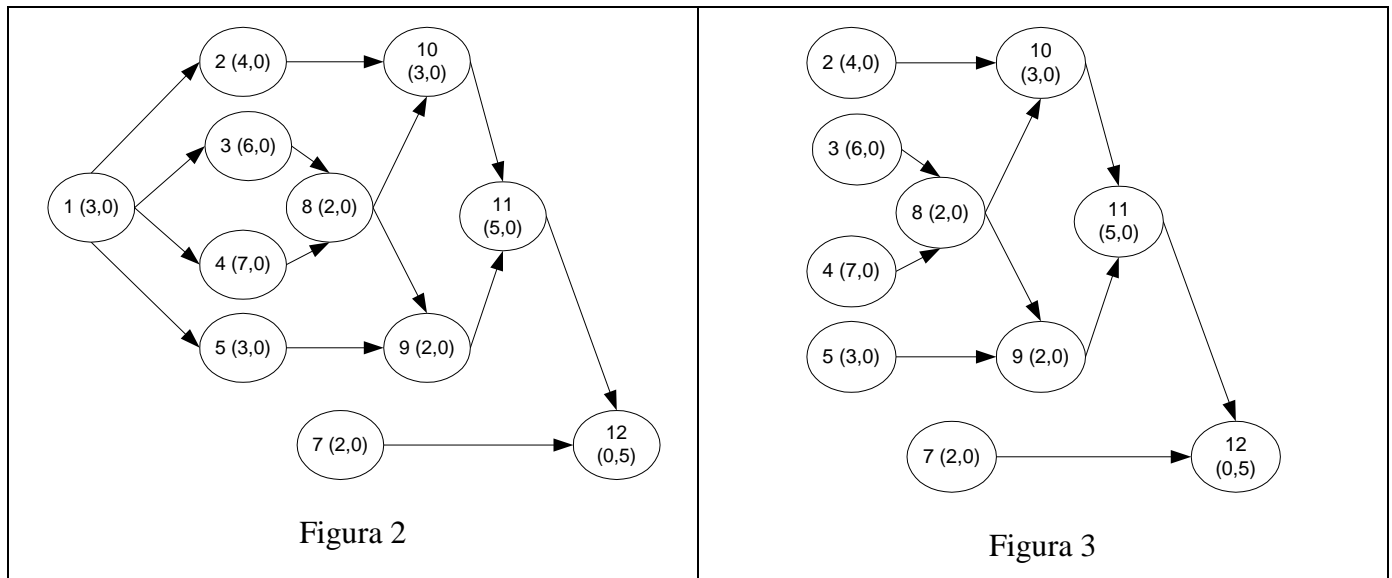
Em um conjunto finito parcialmente ordenado, um elemento é minimal se não tiver predecessores. Em um conjunto finito não vazio parcialmente ordenado, há pelo menos um elemento minimal. Um elemento minimal em um diagrama PERT não tem elementos à sua esquerda.

Para construir uma ordenação topológica, tome um elemento minimal  $x_1$  do conjunto finito parcialmente ordenado  $S$ . Se houver mais de uma possibilidade, pegue qualquer um aleatoriamente. Agora consideremos o conjunto  $S - \{x_1\}$ ; este conjunto ainda é um conjunto parcialmente ordenado; escolha, então, aleatoriamente, um elemento minimal  $x_2$  deste conjunto. Considere agora o conjunto  $S - \{x_1, x_2\}$ ; escolha aleatoriamente um elemento minimal  $x_3$  e prossiga assim por diante, até que todos os elementos tenham sido usados. A ordenação  $x_1 < x_2 < x_3 < \dots < x_n$  é uma ordenação total.

Uma ordenação topológica da ordenação do diagrama PERT na figura 1 é

6,1,7,2,3,5,4,8,10,9,11,12

Neste exemplo, tanto 6 como 1 são minimais e podem ser escolhidos como primeiros elementos. Se o 6 for escolhido e removido do conjunto então, como mostrado na figura 2, o 1 e o 7 serão minimais. Se o 1 for então escolhido e removido da figura (figura 3), então 2, 3, 4, 5 e 7 tornam-se minimais, e qualquer deles pode ser o próximo escolhido. O processo continua até que todos os vértices tenham sido escolhidos. A ordenação topológica fornece uma ordem para realização das tarefas sequencialmente.



O objetivo do trabalho é tomar como entrada um conjunto de tarefas enumerados de 1 a  $n$ , em qualquer ordem, e alocados dinâmica e seqüencialmente na memória. Além disso, tomar um conjunto de pares  $(x,y)$ , significando que a tarefa  $x$  precede a tarefa  $y$ . O número de tarefas  $n$  e o número de pares  $m$  também são fornecidos. O resultado do programa é uma ordenação topológica de tarefas.

### Entrada

A entrada será uma seqüência de tarefas e uma seqüência de pares. A entrada estará em no arquivo texto *ordtopo.in*, começando com um inteiro, que indica a quantidade de  $n$  tarefas no arquivo, seguida pela seqüência de descrições de tarefas. Após isso, tem-se outro número inteiro, que indica a quantidade  $m$  de seqüência de pares.

```

12
Seleção_de_madeira
Entalhamento_dos_arcos
Entalhamento_do_acento
Entalhamento_do_encosto
Entalhamento_dos_braços
Seleção_do_tecido
Costura_da_almofada
Montagem_do_acento_e_do_encosto
Fixação_dos_braços
Fixação_dos_arcos
Aplicação_de_verniz
Colocação_da_almofada
17
1 2
1 3
1 4
1 5
6 7
3 8
4 8
5 9
8 9
2 10
8 10
9 11
10 11
7 12
11 12
1 2
3 1

```

## Saída

Como saída o seu programa deve imprimir no monitor seqüência de tarefas a serem executadas.

```
Tarefas (1,2) ja inseridas !
Tarefas (3,1) formam ciclos !
tarefa 1 : Seleção de madeira
tarefa 6 : Seleção do tecido
tarefa 2 : Entalhamento dos arcos
tarefa 3 : Entalhamento do acento
tarefa 4 : Entalhamento do encosto
tarefa 5 : Entalhamento dos braços
tarefa 7 : Costura da almofada
tarefa 8 : Montagem do acento e do encosto
tarefa 9 : Fixação dos braços
tarefa 10 : Fixação dos arcos
tarefa 11 : Aplicação de verniz
tarefa 12 : Colocação da almofada
```

Além disso, este arquivo deverá apresentar mensagens de erros na validação do arquivo de entrada. Como por exemplo, se existir um par (x,y) na seqüência de pares que já foi cadastrado, então emitir uma mensagem no arquivo de saída informando que o par (x,y) já foi inserido. Outro exemplo, se existir um par (x,y) na seqüência e  $x = y$ , então emitir uma mensagem no arquivo de saída informando que no par (x,y) a tarefa x deve ser diferente da tarefa y. Deve-se verificar, antes de cadastrar o par no diagrama se eles não formam ciclo. Consideremos um conjunto de tarefas  $S = \{x_1, x_2, \dots, x_n\}$ , seja a seguinte seqüência de pares

$x_1 x_2$

$x_2 x_3$

$x_3 x_1 \leftarrow$  Este par forma um ciclo, portanto não pode ser cadastrado no diagrama

## Restrições do projeto

O programa entregue **será avaliado** de acordo com os seguintes itens:

- Funcionamento do programa;
- O programa deve estar na linguagem **C**.
- O quão fiel é o programa quanto à descrição do enunciado;
- Identação, comentários e legibilidade do código;
- Clareza na nomenclatura de variáveis e funções.
- O problema deve utilizar a estrutura **TDigrafo** (conforme visto em sala) armazenando o digrafo em **listas de adjacências**.

## Grupo

A atividade pode ser feita individualmente ou em dupla (somente uma entrega no Moodle com o nome dos dois alunos). Como este trabalho pode ser feito em grupo, evidentemente você pode “discutir” o problema dado com outros grupos, inclusive as “dicas” para chegar às soluções, mas você deve ser responsável pela solução final e pelo desenvolvimento do seu programa. Assim não repasse para e nem copie o programa de outro grupo. Trabalhos considerados plagiados receberão nota 0 (zero).