



**Universidade Federal de São João Del-Rei**

**Ciência da Computação – 1º período**

## **AEDS I**

### **TP1: Dominó e Dengue em Xulams**

**Professor – Leonardo Rocha**

**Isabella Vieira Ferreira**

**(Matrícula: 112050060)**

**<isabellavieira57@yahoo.com.br>**

**Mônica Neli de Resende**

**(Matrícula: 112050045)**

**<monica\_xneli@hotmail.com>**

**03 de junho de 2011**

# **Jogo de Dominó**

## **Índice:**

Introdução.....	4
1- Regras do jogo.....	5
2- Problema proposto.....	5
2.1 Entradas de dados.....	5
2.2 Descrições da solução.....	6
2.3 Processos utilizados.....	7
2.4 Restrições.....	7
2.5 Saída.....	7
3 – Testes e resultados.....	8
4 -Simulação do algoritmo passo-a-passo.....	10
Conclusão.....	15
Referências.....	16

# **Denque em Xulams**

## **Índice:**

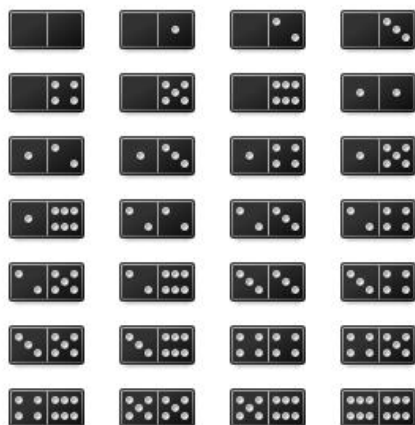
Introdução.....	4
1- Problema proposto.....	11
1.1- Entradas de dados.....	11
1.2- Descrição da solução.....	11
1.3- Processos utilizados.....	12
1.4- Restrições.....	12
1.5- Saída.....	12
2- Testes e resultados .....	13
Conclusão.....	15
Referências.....	16

## Introdução

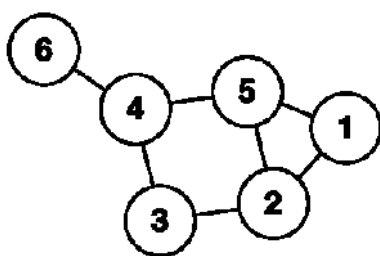
O jogo de dominó aparentemente surgiu na China e sua criação é atribuída a um santo soldado chinês chamado Hung Ming, que viveu de 243 a.C a 182 a.C. O conjunto tradicional de dominós, conhecido como sino-europeu, é formado por 28 peças, ou pedras. Cada face retangular de dominó é dividida em duas partes quadradas, ou "pontas", que são marcadas por um número de pontos de 0 a 6, ou deixadas em branco. O objetivo do jogo é fazer uma união entre todas as peças.

O desafio Dengue em Xulams nos faz preocupar com o surto de Dengue daquele país. Para solucionarmos o problema, deveremos colocar um posto de vacinação em uma vila onde diminua os custos da população com o transporte.

“Os jogos estabelecem uma forma de atividade do ser humano, tanto no sentido de entreter e de aperfeiçoar ao mesmo tempo, quanto possibilitam desenvolver uma série de desempenhos em um contexto educacional como raciocínio lógico e aritmético dos aprendizes.”



*Exemplo de jogo de dominó de 28 peças*



*Exemplo de vilas interligadas por linhas de trem*

# Jogo de Dominó

## 1. Regras do jogo

As pedras são denominadas de acordo com os números em suas extremidades. Os números variam de 0 a 6. Cada número deve “encaixar” em outro dominó de extremidade igual, para que se obtenha uma combinação válida. Se sobrar peças que não encaixam umas nas outras, teremos então uma combinação inválida.



*Combinação válida de dominós*

## 2. Problema proposto

O desafio é escrever um programa que determine se é possível organizar todas as peças recebidas em uma sequência válida, obedecendo às regras do jogo de dominó.

### 2.1 Entradas de dados

Deve-se inserir a combinação que se deseja processar em um arquivo texto, com o nome **‘domino.txt’** (exatamente como foi exposto). O arquivo deve estar na mesma pasta do código-fonte (.pas), no seguinte formato:

## Exemplo de Entrada

```
3
0 1
2 1
2 1
```

Note que o primeiro número indica a quantidade de dominós e o número 0 implica no final do arquivo de entrada. Se o arquivo de entrada não estiver de acordo com as especificações acima, o resultado obtido não será de acordo com o esperado.

## 2.2 Descrições da Solução

O algoritmo desenvolvido inicia com a leitura do arquivo texto, atribuindo cada extremidade do dominó a uma variável unidimensional homogênea (vetor principal).

0	1	2	1	2	1
---	---	---	---	---	---

*Figura correspondente ao exemplo de entrada anterior*

O primeiro passo é verificar se o primeiro dominó (uma das duas extremidades) se encaixa com outro dominó do vetor principal, atribuindo as extremidades da formação obtida a duas variáveis (extremidades principais) e assinalando os dominós já usados. A partir disso, todos os outros dominós serão comparados primeiramente com as extremidades principais, atualizando-as se for uma combinação válida, e caso não ocorra uma formação, são comparados com um vetor denominado “sobrou” (responsável por armazenar os dominós ou as extremidades de alguma formação que não se encaixou com as extremidades principais), do contrário, são comparados com o vetor principal.

A partir do momento que há algum dominó no vetor “sobrou”, estes são comparados com as extremidades principais frequentemente. Segue-se assim até que todos os elementos do vetor principal estejam todos usados.

Para definirmos se a sequência é válida, consideremos que o vetor “sobrou”, esteja vazio ou assinalado como já usado, caso contrário, será uma combinação inválida.

## 2.3 Processos utilizados

Utilizamos os conceitos de par e ímpar para identificar uma peça de dominó no vetor. Se a posição no vetor for um número par, então a extremidade deste dominó é a posição anterior. Caso contrário, será a próxima posição no vetor.

Da seguinte forma:

1	2	3	4	5	6	 Índices do vetor
0	1	2	1	2	1	

*Figura correspondente ao exemplo de entrada anterior*

Para referirmos que o vetor principal e o vetor “sobrou” está vazio, os inicializamos com -3. E para distinguir um dominó já utilizado, ele será substituído por -1.

## 2.4 Restrições

- O programa não comporta entrada de números negativos.
- O programa comporta no máximo 50 dominós em cada teste.

## 2.5 Saída

As saídas serão constituídas pela palavra **Teste** seguidas do número referente ao mesmo. Na linha seguinte será impresso o resultado do teste, da seguinte forma: **sim**, para teste válido, e **nao**, para teste inválido (a ausência do acento gráfico deve-se ao fato das restrições da linguagem de programação).

## Exemplo de Saída

```
Teste 1
sim
```

*Saída correspondente ao exemplo de entrada anterior*

### 3. Testes e resultados

O código foi gerado pelo Pascalzim 5.1.0.

Os testes foram realizados em uma máquina com a seguinte descrição:

Hardware/Software	Modelo
<b>Sistema operacional:</b>	Windows 7 Ultimate 32 bits e Ubuntu 10.04
<b>Processador:</b>	Intel Core 2 Duo CPU T6400 2.00 GHz
<b>RAM:</b>	3,00 GB, DDR2, 667MHz

*Testes:*

Entrada	Saída
3	Teste 1 sim
0 1	
2 1	
2 1 0	

Entrada	Saída
6	Teste 2 sim
3 0	
0 0	
1 6	
4 1	
0 6	
2 3	
0	



Entrada	Saída
2	Teste 3 nao
1 1	
0 0	
0	

Entrada	Saída
10	Teste 4 nao
1 0	
3 2	
5 6	
4 3	
3 0	
4 1	
0 0	
2 1	
3 5	
1 2	
0	

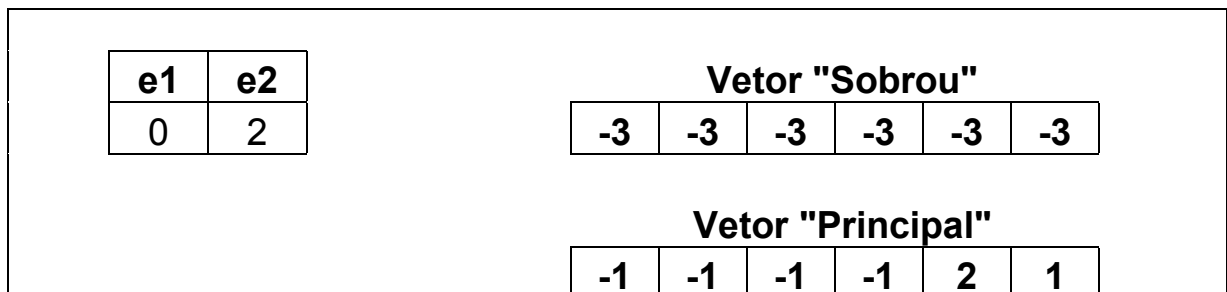
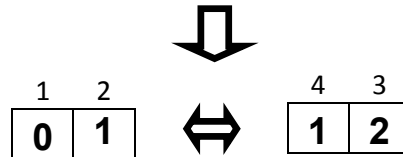
Entrada	Saída
8	Teste 5 sim
1 2	
4 2	
5 4	
0 5	
6 3	
3 1	
1 0	
1 6	
0	

Entrada	Saída
8	Teste 6 nao
1 2	
5 4	
4 0	
3 5	
6 1	
1 5	
2 4	
5 1	
0	

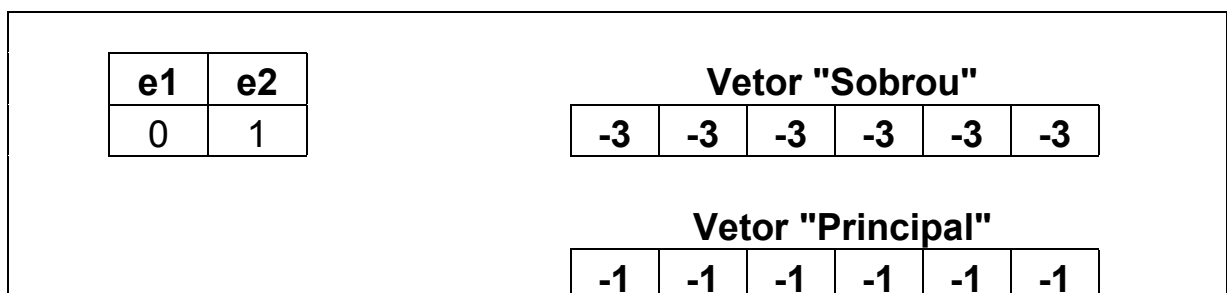
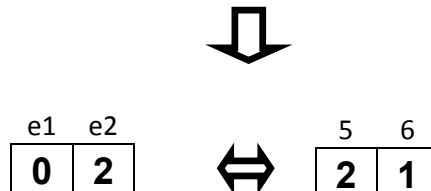
#### 4.Simulação do algoritmo passo - a - passo

1	2	3	4	5	6	Índices do vetor
0	1	2	1	2	1	Sequência de dominós lida do arquivo

Juntando as peças de dominós semelhantes



Juntando as peças de dominós semelhantes



Teste 1

sim

# Denque em Xulambs

## 1. Problema proposto

Dado um país com várias vilas interligadas por uma rede de linhas de trem, a qual foi atacada por surto de dengue, o desafio é verificar em qual vila deverá ficar um posto de saúde para a vacinação da população do país em questão, de modo que, os gastos com o transporte seja o menor possível.

### 1.1 Entrada de dados

Devem-se inserir as ligações (arestas do grafo) entre as vilas em um arquivo texto, com o nome '**xulambs.txt**' (exatamente como foi exposto). O arquivo deve estar na mesma pasta do código-fonte (.pas), no seguinte formato:

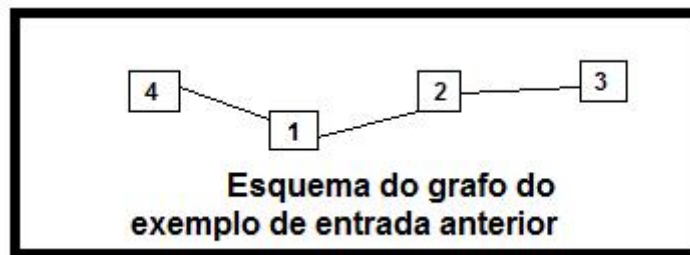
#### Exemplo de entrada

```
4
1 2
2 3
1 4
0
```

Note que o primeiro número indica a quantidade de vilas do país seguido de (vilas-1) linhas, e o número 0 implica no final do arquivo de entrada. Se o arquivo de entrada não estiver de acordo com as especificações acima, o resultado obtido não será de acordo com o esperado.

### 1.2 Descrição da solução

O algoritmo desenvolvido inicia com a leitura do arquivo texto, atribuindo cada ligação (aresta) das vilas a uma variável bidimensional homogênea (matriz – de tamanho: (nº de vilas -1) x 3). Quando as vilas que possuem somente uma aresta são “cortadas” é atribuindo uma *flag* a 3ª coluna da matriz.



Matriz		
1	2	
2	3	
1	4	

*Figura correspondente ao exemplo de entrada anterior*

### 1.3 Processos utilizados

Para identificar quais vilas foram “cortadas” atribuímos -2 a matriz, onde este pode ser utilizado para contar quantas vezes cada número se repete. À medida que são contados, o ‘-2’ é substituído por ‘-1’ para que este não seja mais usado.

### 1.4 Restrições

- O programa comporta no máximo 100 vilas em cada teste.
- Caso haja duas vilas possíveis para colocar o posto de vacinação, para manter um padrão, definiremos que a primeira será a resposta.

### 1.5 Saídas

As saídas serão constituídas pela palavra **Teste** seguidas do número referente ao mesmo. Na linha seguinte será impresso o número da vila na qual deve ser instalado o posto de vacinação

<p>Teste 1</p> <p>1</p>
-------------------------

## 2. Testes e resultados

O código foi gerado pelo Pascalzim 5.1.0.

Os testes foram realizados em uma máquina com a seguinte descrição:

Hardware/Software	Modelo
Sistema operacional:	Windows 7 Ultimate 32 bits e Ubuntu 10.04
Processador:	Intel Core 2 Duo CPU T6400 2.00 GHz
RAM:	3,00 GB, DDR2, 667MHz

*Testes:*

Entrada	Saída
10	Teste 1 1
1 2	
1 3	
2 4	
2 5	
3 6	
4 7	
5 8	
6 9	
9 10	
0	

Entrada	Saída
5	Teste 2 2
1 2	
2 3	
2 4	
3 5	
0	

Entrada	Saída
18	Teste 3 15
1 2	
2 6	
3 5	
4 5	
5 6	
8 7	
10 7	
7 9	
6 9	
9 15	
16 14	
15 14	
14 13	
13 17	
11 13	
12 13	
18 12 0	

Entrada	Saída
7	Teste 4 7
1 2	
2 5	
7 4	
7 2	
4 6	
3 4	
0	

## Conclusão

Neste trabalho aprofundamos ainda mais nossos conhecimentos sobre programação.

Por meio deste, conseguimos perceber que para solucionar um problema é preciso conhecer técnicas como estruturas condicionais, estruturas de repetição, matriz e vetor.

Conforme proposto, implementamos esse trabalho utilizando leitura em arquivo.

No jogo de dominó, utilizamos séries de comparações para chegar ao resultado, e no algoritmo Dengue em Xulams utilizamos o conceito de grafo, da forma que a medida que as vilas estivessem nas extremidades, eles seriam “cortadas”.

Usando todos os conhecimentos vistos em sala de aula e adequando ao desafio proposto, pudemos fazer com que o usuário se interaja com o computador de forma a resolver problemas cotidianos de forma eficiente.

## Referências

[1] Projeto de Algoritmos - Nivio Ziviani

[2] <http://educador.brasilecola.com/trabalhodocente/jogodedomino.htm>

*Autora:* Amelia Hamze

Profª UNIFEB/CETEC e FISO - Barretos

Colunista Brasil Escola

[3] <http://pt.wikipedia.org/wiki/Dominó>