

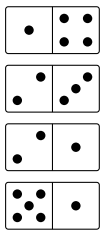
Universidade Federal do ABC  
Técnicas Avançadas de Programação – 2018.Q1  
Profs. Daniel M. Martin e Francisco Isidro Massetto

Projeto 2: Trilha de Dominós!  
Valor: 10 pontos. Entrega: 23/03

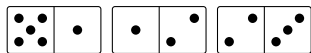
## 1 Descrição

Todos já brincaram de jogar Dominó alguma vez na vida. As peças são retangulares e, cada uma delas, possui dois números, um em cada extremidade do retângulo. Neste exercício-programa também vamos lidar com peças de dominó.

Sua tarefa consiste em fazer um programa que recebe dois inteiros  $n \geq k > 0$  e  $n$  peças de dominó  $(a_1, b_1), (a_2, b_2), \dots, (a_k, b_k)$ , com  $a_i, b_i \in \mathbb{N}$  para todo  $i \in \{1, \dots, k\}$ , e que decide se é possível colocar  $k$  dessas peças numa fileira de modo que peças consecutivas se casam como num jogo de dominó. Por exemplo, considere a instância  $n = 4, k = 3$  onde as quatro peças de dominó são:



Nesse caso, é possível colocar três dessas peças numa fila de modo que peças consecutivas se casem:



Note que as peças podem ser “rotacionadas” de 180 graus se for necessário. Note também que, se tivéssemos  $k = 4$ , a instância do problema não teria solução.

## 2 Tarefa

Seu programa deve:

- (a) (8 pontos) decidir se existe solução e, em caso positivo, exibir uma solução na tela;
- (b) (4 pontos) enumerar todas as soluções na tela;
- (c) (4 pontos) enumerar todas as soluções na tela sem repetições.

## 3 Conceitos abordados

### 3.1 Backtracking

O principal intuito deste exercício-programa é incentivar a prática da técnica de programação chamada Backtracking.

## 4 Especificações

### 4.1 Entrada e saída

Os dados serão lidos pelo programa da entrada padrão<sup>1</sup>. A primeira linha da entrada terá os inteiros  $n$  e  $k$  separados por espaço em branco. As  $n$  linhas seguintes irão conter informações das peças de dominó. A  $i$ -ésima dessas  $n$  linhas terá o par de inteiros não negativos  $a_i$  e  $b_i$ , separados por espaço em branco.

A saída do programa deve apresentar uma solução por linha, podendo ser vazia se a instância não admitir solução. No caso de existir solução, cada linha da saída irá representar uma solução por meio de  $2k$  números inteiros não negativos separados por espaços em branco que, aos pares, indicam as peças usadas naquela solução.

Por exemplo, considerando a instância da Seção 1, um programa que só implemente o item (a) poderia ter como saída uma única linha como abaixo.

```
5 1 1 2 2 3
```

Se o programa implementar até o item (c), a saída correta para aquela instância seria:

```
4 1 1 2 2 3
3 2 2 1 1 4
3 2 2 1 1 5
5 1 1 2 2 3
```

### 4.2 Script de correção

Será divulgado em breve no Tidia.

---

<sup>1</sup>Se você conseguiu a proeza de chegar a este curso sem saber o que é *entrada padrão* ou *saída padrão*, pesquise sobre isso e aprenda esses conceitos antes de fazer qualquer outra coisa! Dica: digite *standard input output* no Google.