

# 3ª Prova de Algoritmos e Estruturas de Dados I

11/12/2013

## Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.  
R: A compreensão do enunciado faz parte da prova.
- P: Se eu consultar algum material próprio ou de algum colega, o que acontecerá comigo?  
R: A prova é individual e sem consulta. Qualquer tentativa de fraude acarretará abertura de processo administrativo na UFPR.
- P: Posso entregar a prova a lápis?  
R: Uma prova é um documento, portanto deve ser entregue a caneta.
- P: O que será corrigido?  
R: A lógica, a criatividade, a sintaxe, o uso correto dos comandos, a correta declaração dos tipos, os nomes das variáveis, a indentação, o uso equilibrado de comentários no código e, evidentemente, a clareza. A modularidade, o correto uso de funções e procedimentos, incluindo passagem de parâmetros e o bom uso de variáveis locais e globais serão especialmente observados.

## Preâmbulo

*Polling* é uma técnica utilizada em Redes Neurais Convolucionais (CNN's) que reduz a dimensionalidade da imagem visando agrupar características similares em uma única informação. A funcionalidade é similar a convolução, porém mais simples.

Uma variação é a *MaxPooling* que aplica um filtro de máxima para cada pedaço da imagem. A Figura 1 ilustra o processo. Neste caso o passo aplicado foi de tamanho 2, ou seja, é analisada uma área  $2 \times 2$  que é saltada e nenhum elemento é novamente analisado. Uma outra possibilidade é que o passo seja de tamanho 1, assim a dimensão da imagem é reduzida somente em uma linha e uma coluna (ver exemplo de saída 2).

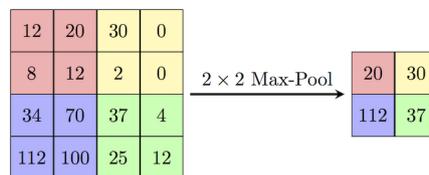


Figura 1: Funcionamento do MaxPooling

## Questão única (valor 100 pontos.)

Faça um programa em *Free Pascal* que leia um tamanho  $N$ ,  $0 < N \leq 1024$ , sendo  $N$  um número par e lê também qual o tamanho passo (1 ou 2). E que leia uma matriz quadrada  $N \times N$  de números inteiros. O programa realiza uma operação de *MaxPooling* com o passo definido na matriz lida e imprime a matriz resultante. Montar as funções e/ou procedimentos necessários para a execução do programa.

Exemplo1:

```
4
2
12 20 30 0
 8 12  2 0
34 70 37 4
112 100 25 12
```

Saída:

20 30  
112 37

Exemplo2:

4  
1  
12 20 30 0  
8 12 2 0  
34 70 37 4  
112 100 25 12

Saída:

20 30 30  
70 70 37  
112 100 37