

# Exercício para Casa — Algoritmo de Warshall

O famoso algoritmo de Warshall calcula o fecho transitivo de um grafo sem pesos, representado por uma matriz (Booleana) de adjacências. Trata-se de um algoritmo amplamente documentado, por exemplo nestas notas:

<https://cs.winona.edu/lin/cs440/ch08-2.pdf>

O exercício, para resolver até à data do teste, consiste nas duas seguintes tarefas, tomando por base a função anotada fornecida em baixo :

## 1. Verificação básica.

Começando por remover a pós-condição e o `assert` na função `WarshallTC`, anotar a função com todos elementos necessários (possivelmente modificando a pré-condição especificada) para, com recurso aos plugins RTE e WP do Frama-C:

- a. Provar a sua *segurança de execução* (operações de memória e aritméticas).
- b. Provar a *frame condition* contida na sua especificação.
- c. Provar a *terminação* da função.

## 2. Verificação intermédia.

Repondo agora a pós-condição e o `assert`, anotar a função `WarshallTC` com todos os elementos necessários para:

- a. Provar o `assert`, que exprime que a matriz `R` é inicializada correctamente.
- b. Provar a pós-condição, que exprime que a matriz `A` não é alterada.

Depois de resolvidas, deverá submetê-las seguindo as instruções que serão disponibilizadas no Blackboard.

```

1  #define MAXVERTICES 10
2  typedef int Graph[MAXVERTICES][MAXVERTICES];
3
4  /*@ requires ... &&
5      @      \valid(A+(0..(n*n-1))) &&
6      @      \valid(R+(0..(n*n-1))) &&
7      @      \separated(A+(0..n*n-1), R+(0..n*n-1)) ;
8      @ assigns R[0..n-1][0..n-1];
9      @ ensures \forall integer k, l;
10     @      0 <= k < n && 0 <= l < n ==>
11     @      A[k][l] == \at(A[k][l],old);
12     @*/
13 void WarshallTC (Graph A, Graph R, int n) {
14     int i, j, k;
15
16     for (i=0 ; i<n ; i++)
17         for (j=0 ; j<n ; j++)
18             R[i][j] = A[i][j];
19
20     /*@ assert \forall integer k, l;
21         @      0 <= k < n && 0 <= l < n ==> R[k][l] == A[k][l];
22         @*/
23
24     for (k=0 ; k<n; k++)
25         for (i=0 ; i<n ; i++)
26             for (j=0 ; j<n ; j++)
27                 if (R[i][k] && R[k][j])
28                     R[i][j] = 1;
29
30 }

```