Homework Assignment 3

Given 11/18/2014, due 12/03/2014

Write code that finds a maximum flow in a directed graph, using the Ford-Fulkerson algorithm on capacities given as matrix

`void maximum_flow(int n, int s, int t, int *capacity, int *flow)`

Your function has the following arguments:
  - `n`: the number of vertices of the graph,
  - `s`: the start vertex,
  - `t`: the target vertex
  - `capacity`: the matrix of edge capacities.
  - `flow`: the matrix used to return the maximum flow.

The vertices are numbered from 0 to `n-1`, so `s` and `t` are numbers in that range.

`capacity, flow` are a pointers to $n \times n$ matrices of nonnegtive integers; the array element `capacity[i][j]` is the capacity of the edge from `i` to `j`, and can be accessed as `*(capacity + i*n + j)`. Your function should return in the matrix `flow` the flow values of the maximum flow from `s` to `t`. The `flow` variable of your function points to space allocated for the `flow` matrix.

Your function will need the following auxiliary arrays:
  - an $n \times n$ matrix to hold the current flow,
  - an $n \times n$ matrix to hold the current residual capacities,
  - an array to maintain which vertices are already visited in the search of an augmenting path from `s` to `t` with positive residual capacity.

You have to allocate the auxiliary arrays. You can use either BFS or DFS for the search of the augmenting path.