Exercise 12.1  Matchings.

Let $G = (V, E)$ be an undirected graph. A set $M \subseteq E$ is called a matching if any two different edges $e, e' \in M$ do not share a common vertex. The size of a matching is the number of its edges. A matching $M \subseteq E$ is called maximal if there is no edge $e \in E \setminus M$ such that $M \cup \{e\}$ is a matching. A matching of size $\lceil |V|/2 \rceil$ is called perfect.

a) Let $M^*$ a matching of maximum cardinality. Show that every maximal matching $M \subseteq E$ satisfies $|M| \leq |M^*| \leq 2|M|$.

b) Provide a graph with 8 vertices and a maximal matching that has exactly half the number of edges that a matching of maximum cardinality has.

c) Describe an algorithm that computes a maximal matching as efficient as possible. Provide also the running time.

d) Provide a subset of vertices of the following bipartite graph that using Hall’s theorem prove that the graph has no perfect matching.

Exercise 12.2  Dominos.

The goal in this exercise is to cover a board with a grid consisting of $n$ rows and $m$ columns with a maximal number of dominos. However, some cells in the grid need to remain uncovered. The grid is given by a $(n \times m)$ matrix of zeros and ones, where one means that the corresponding grid cell needs to remain uncovered. Calculate the maximum number of dominos that can be placed on the grid without overlapping and without covering the grid cells containing a one in the matrix. The following example shows a grid with three rows and six columns. The black cells should remain uncovered. One can place at most six dominos on this grid.
Exercise 12.3 Programming exercise: Dominoes.

In this exercise we implement the solution of exercise 12.4. You are given an \((n \times m)\) matrix of zeros and ones, where one means that the cell of a matrix is occupied. Your goal is to calculate maximum amount of disjoint dominoes \((1 \times 2\) or \(2 \times 1\) matrices) that you can put in this matrix without covering cells with a one in the matrix.

**Input** The first line contains only the number \(t\) of test instances. After that, we have the following description of test instance. First come dimensions \(n, m \leq 30\), then there come \(n\) lines each containing \(m\) symbols 0’s and 1’s.

**Output** For every test instance we output only one line. This line contains the value of a maximum number of dominoes you can put in a given matrix.

**Example**

*Input:*

\[
3 \\
1 3 \\
0 1 0 \\
3 2 \\
1 0 \\
0 0 \\
0 1 \\
2 3 \\
1 0 1 \\
0 0 0
\]

*Output:*

\[
0 \\
2 \\
1
\]

**Remarks**

- In 30% of the testcases we have \(n, m \leq 5\), in 80% of the testcases \(n, m \leq 10\).

- \(O((nm)^2)\) running time is enough to get 100% of points.

**Hand-in:** until Wednesday, 20th May 2015.