MA3077 Tutorial 6 (24 Oct 2023)

Lecture 15 self-study

```
function y = is_stable(A)

% idea
% compute A- := max_i min_j a_{ij} and A+ := min_j max_i a_{ij}
% if A- == A+, return true; otherwise, return false

A_minus = max(min(A,[],2),[],1);
                              % max(min(A,[],2)) would also work
A_plus = min(max(A,[],1),[],2);
                              % min(max(A,[],1)) would also work
y = (A_minus == A_plus);

end
```

Lecture 16 self-study

```
% to learn about linprog, type <help linprog> or <doc linprog>
in the command window.
% Note that linprog solves minimization problems only!

% set the problem specs
f = -[0;0;0;1];                % objective function to minimize
A = -[0 5 2 -1; -2 4 3 -1; 2 -3 -4 -1; 1 0 0 0; 0 1 0 0; 0 0 1
0];                            % inequality constraints matrix
b = [0 0 0 0 0 0];             % inequality constraints rhs
Aeq = [1 1 1 0];               % equality constraints matrix
beq = [1];                     % equality constraints rhs

% call linprog
[x,fval,exitflag,output] = linprog(f,A,b,Aeq,beq);

% print optimal solution and value
fprintf('%.4f ',x,fval);
```

Further exercise

- (Suggested by one of your classmates) Instead of incorporating the non-negativity constraint $x \geq 0$ into the lower half of the inequality constraints matrix A, it would be simpler to set this as a lower bound and call `linprog(f,A,b,Aeq,beq,lb,ub)`. In order to specify `lb`, one needs a lower bound for v as well. Can you think of one? (Hint: What is the *worst possible* payoff for Player 1?)
- Compute the optimal mixed strategy for Player 2, and verify that it leads to the same expected outcome.
- (Challenging) Write a MATLAB program that takes as input *any* payoff matrix A.