last lecture: using LP to find matrix game value

- our story so far ...
- classical GT: matrix games
- Von Neuman's theorem: every matrix game has a value ...
- today: how to use linear programming to find that value
- lecture assumes you have read and understood text chapter 5.1
- details come out of Appendix B

- example matrix game from Ch. 5

```
      C            what happens with   pure strat   r1 vs c1 ?

    2  -1                                           r2 vs c1 ?

 R  1   2                                           r1 vs c2 ?

                                                    r2 vs c2 ?


now consider R plays mixed strategy  x * r1 + y * r2,

   where x,y are probabilities (non-negative rationals, x + y = 1)


     case 1) x * r1 + y * r2 versus c1 ?

     case 2) x * r1 + y * r2 versus c2 ?


     case 1) R exp. payout  x * 2 + y * 1

     case 2) R exp. payout  x *-1 + y * 2
```

- Rose wants to maximize her guaranteed expected payout
- for any mixed Rose-strat (x, y), we assume Colin will play the minimizing pure C-strat
- thus Rose wants to maximize `min{ 2x + y, -x + 2y }`
- Rose wants to

$$\text{maximize } z \text{ such that}$$

$$z \le 2x + y$$

$$z \le -x + 2y$$

$$0 \le x, y \le 1$$

$$x + y = 1$$

- evaluate this program at `https://sagecell.sagemath.org/`

```
p = MixedIntegerLinearProgram()
v = p.new_variable(real=True, nonnegative=False)
x, y, z = v['x'], v['y'], v['z']
p.set_objective(z)
p.add_constraint(z <= 2*x +   y)
p.add_constraint(z <=  -x + 2*y)
p.add_constraint(x + y == 1)
p.add_constraint(x >= 0)
p.add_constraint(y >= 0)
p.solve()
p.get_values(z,x,y)
```

- you should get this output

```
[1.25, 0.25, 0.75]
```

- this tells us that R is guaranteed expected payout 1.25 when she plays row 1 with probability .25 and row 2 with probability .75
- check this: her expected payout against pure strat column 1?   column 2?
- can we verify that this is her maximum guaranteed expected payout?
- yes: Von Neumann's theorem, which tells us that there will be a mixed strategy for Colin with guaranteed expected payout (owing to Rose) at most 1.25
- let's use the same method as above to find an optimizing mixed strategy for Colin

- Colin wants to minimize his guaranteed expected payout
- for any mixed Colin-strat (s,t) we assume Rose will play the maximizing pure R-strat
- Colin wants to minimize `max{ 2s - t, s + 2t }`
- reformulate this as a maximization problem (for SageMath)
- Colin wants to maximize `min{ -2s + t, -s - 2t }`

maximize z such that

$$z \leq -2s + t$$

$$z \leq -s - 2t$$

$$0 \leq s, t \leq 1$$

$$s + t = 1$$

- evaluate this program at `https://sagecell.sagemath.org/`

```
p = MixedIntegerLinearProgram()
v = p.new_variable(real=True, nonnegative=False)
s, t, z = v['s'], v['t'], v['z']
p.set_objective(z)
p.add_constraint(z <= -2*s + t)
p.add_constraint(z <=  -s - 2*t)
p.add_constraint(s + t == 1)
p.add_constraint(s >= 0)
p.add_constraint(t >= 0)
p.solve()
p.get_values(z,s,t)
```

- you should get this output

```
[-1.25, 0.75, 0.25]
```

- this tells us that C is guaranteed expected payout -1.25 when he plays col 1 with prob .75 and col 2 with prob .25
- check this: his expected payout against pure strat row 1?   row 2?
- can we verify that this is his guaranteed expected payout?
- yes, because R has a guaranteed expected payout exactly the negative of this amount
- we have found, and verified, that value for this game is 1.25
  - in expected value, by following her mixed (.25, .75) strategy, R is guaranteed to win at least this amount against any pure col strat
  - in expected value, by following his mixed (.75, .25) strategy, C is guaranteed to lose at most this amount against any pure row strat

- another example

		C	
	1	2	1
R	1	0	2
	3	1	0

- R wants to maximize her guaranteed expected payout
- for any mixed R-strat (a,b,c), assume C plays the minimizing pure C-strat
- R wants to maximize `min{ a + b + 3c, 2a + c, a + 2b}`
- R wants to

$$\text{maximize } z \text{ such that}$$

$$z \leq a + b + 3c$$

$$z \leq 2a + c$$

$$z \leq a + 2b$$

$$0 \leq a, b, c \leq 1$$

$$a + b + c = 1$$

- use sagemath to find
  - value of this matrix game
  - a minimax strategy for R
  - a minimax strategy for C