Midterm 2

6:00-8:00pm, 13 April

Notes: There are **four** *questions on this midterm. Answer each question part in the space below it, using the back of the sheet to continue your answer if necessary. If you need more space, use the blank sheet at the end.* **None of the questions requires a very long answer, so avoid writing too much! Unclear or long-winded solutions will be penalized.** *The questions vary quite a bit in difficulty, so if you are having problems with part of a question, leave it and try the next one. The approximate credit for each question part is shown in the margin (total 65 points). Points are not necessarily an indication of difficulty!*

Your Name: **Your Section No:**

---

1. True or False?

For each of the following statements, say whether the statement is True or False. If it is True, give a **brief** explanation (≈ one sentence); if it is False, give a **simple** counterexample.

(i) If all the capacities of a flow network are multiplied by a positive number λ, then the value of the maximum flow is multiplied by exactly λ. *3pts*

(ii) If a positive number λ is added to all the capacities of a flow network, then the value of the maximum flow is increased by exactly λ. *3pts*

(iii) If the optimal value of a linear program is unbounded, then the dual program has no feasible solution. *3pts*

[continued on next page]

[Q1 continued]

(iv) Consider the two-person zero-sum game defined by the following matrix. (As usual, the matrix entries correspond to the gain for the Row player.) *3pts*

	a	b
A	4	-2
B	0	1

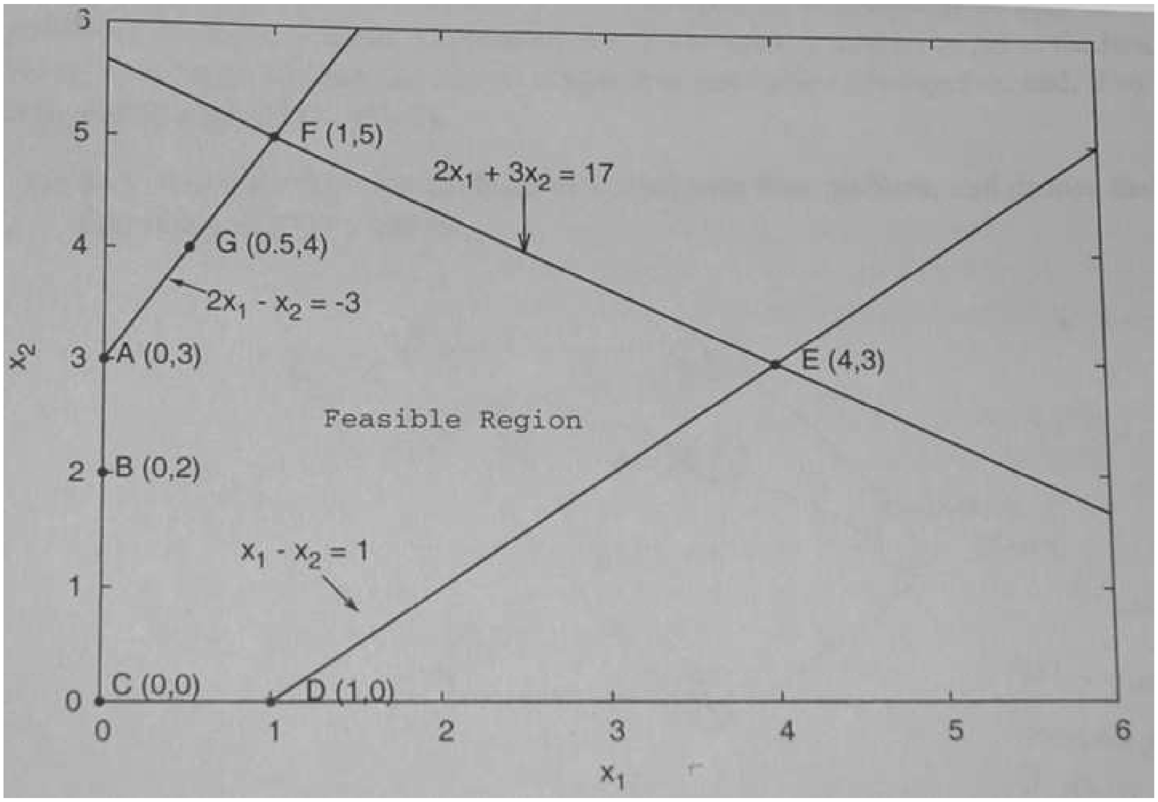
Given that the value of this game is 4/7, the mixed strategy (1/7, 6/7) (i.e., play *A* with probability 1/7 and *B* with probability 6/7) is optimal for the Row player.

(v) In the game from part (iv), suppose that the Column player announces that he will play the mixed strategy (1/3, 2/3). Then the above strategy (1/7, 6/7) is an optimal response for the Row player. *3pts*

[continued on next page]

2. Linear Programming

This problem concerns the following figure, which shows the feasible region of a linear program in two dimensions.



(a) Write down the set of constraints for this linear program. *5pts*

(b) Suppose now that our goal is to maximize the objective function $2x_2$-$3x_1$. Assuming that the simplex algorithm starts at vertex D, write down a possible sequence of vertices that it will visit before it terminates. Also, state the optimal value of the objective function, and the corresponding values of the variables at the optimum. *5pts*

(c) Write down the dual of the above linear program. *5pts*

[continued on next page]

3. Happy families

A group of n families wish to travel together to the beach. Between them, they have a total of m vehicles. The number of people in family i is f_i, while the number of people (including the driver) who can ride in vehicle j is v_j, for $1 \leq i \leq n$ and $1 \leq j \leq m$. There is no constraint on who drives each vehicle. The problem is to decide whether it is possible for all the family members to get to the beach in a single run of the vehicles, in such a way that no two members of any family ride together, and, if so, to come up with an assignment of people to vehicles.

(a) Show how to reduce this problem to a maximum flow problem, and deduce that it can be solved in time polynomial in n and m. *10pt*

(b) Explain why we may assume that the maximum flow problem has an *integer* solution, and also why this is important in your reduction. *5pts*

[continued on next page]

4. Running a simulation on multiple machines

You want to run a long, compute-intensive simulation program, and you have access to a network of *m* computers. Some other computations are already scheduled on these machines, so at any given time the amount of resources available to you on each machine is limited. For each machine *j*, $1 \leq j \leq m$, you are given a sequence $(a_1^j, \ldots, a_n^j)$ of non-negative integers, where a_t^j specifies how many cycles of your simulation can be performed on machine *j* in minute *t*. At the end of each minute, you are free to either continue on your current machine or to switch to another machine. However, there is a penalty for switching: if you switch in minute *t*, then in that minute you are only able to perform *half* of the available cycles on the new machine (with fractions rounded down). Your task is to come up with a schedule that maximizes the total number of simulation cycles performed in *n* minutes. Your schedule will be of the form $(s_1, \ldots, s_n)$, where $s_t \in \{1, \ldots, m\}$ denotes the machine used in minute *t*. For example, if there are two machines with sequences (10, 1, 1, 30) and (5, 5, 20, 20) respectively, the optimal schedule is (1, 2, 2, 2) and achieves a total of 10 + 2 + 20 + 20 = 52 cycles.

(a) Here is a "greedy" algorithm for this problem. Start on a machine with largest value a_1^j. Suppose we have already decided the schedule for the first $t-1$ minutes, and suppose the current machine is *j*. If $a_t^j \geq \lfloor a_t^k/2 \rfloor$ for all $k \neq j$, then continue on machine *j*, else switch to the machine *k* with the largest value of a_t^k. (Thus we switch machines if and only if we can perform more cycles in minute *t* by switching.) Give a *simple* example (with $m = 2$ machines and $n = 3$ minutes)that shows that this greedy approach does not always produce an optimal schedule. *5pts*

(b) For $1 \leq t \leq n$ and $1 \leq j \leq m$, let $C(t, j)$ be the maximum number of cycles that can be performed in the first *t* minutes, assuming that $s_t = j$ (i.e., the schedule ends on machine *j*). Write down the recurrence relation satisfied by $c(t, j)$. Don't forget to include the base cases! *6pts*

[continued on next page]

[Q4 continued]

(c) Use part (b) to design a dynamic programming algorithm that computes the maximum number of cycles that can be performed in n minutes. *3pts*

(d) What is the running time of your algorithm, as a function of n and m? Justify your answer. *3pts*

[The end]