Econ 172A, Fall 2007: Final Examination Answers

Grading Notes

1. I propose to give two points for each correct route, one point for each correct cost, and six points for a justification. Students who use the correct method but make mistakes due to a careless error (for example, an error in addition), should not be punished repeatedly. (So perhaps the formula above is too mechanical.)

2. I would give no credit unless the answer is a spanning tree. Give six points for the correct tree and two more for the cost (subtract one for an obvious addition error) and four more for a coherent explanation of what they did.

3. Give 5 points if the student formulates the problem as a max flow problem and 3 more points if he or she does so correctly. The only way I know how to show that the original problem is not feasible is to show that the maximum flow does not meet demand. If they can come up with a cut that proves this they get 15 more points. Seven points for the second part of the problem. The second part of the problem has many correct answers and students should be able to get these points even if they do not recognize the problem as a maximum flow problem. Students who begin the algorithm correctly but stop too early or make conceptual mistakes should earn a part of the fifteen points. I am not sure how to allocate partial credit.

4. They need to come up with a lower bound ($-\infty$ is ok) (3 points) and an upper bound (here they must solve a relaxed problem somehow (6 points). They must pick some variable to branch on, identify the possible values that the variable can take (4 points), solve the associated problem (they are so simple that they can solve them either in relaxed form or in integers) (5 points), interpret the results (by proving new bounds) (5 points), and identify a solution (3 points).

5. They must provide an example, an argument about what happens under the greedy algorithm, and they should exhibit the solution to the problem and show that it is different from what the greedy algorithm provides. Give three points for any valid knapsack example. Five more points for stating what the greedy algorithm does or solving the problem. Seven more points for completing the argument. This way, a student who writes down a knapsack problem and solves it gets 8 points even if the problem does not have the desired property.

6. Straightforward.

1.

Iteration	1	2	3	4	5	6	7	8
1	0*	294	∞	227	206**	335	∞	∞
2	0*	294	∞	227**	206*	335	417	633
3	0*	294	∞	227*	206*	335	280**	633
4	0*	294**	∞	227*	206*	335	280*	633
5	0*	294*	724	227*	206*	318**	280*	633
6	0*	294*	721	227*	206*	318*	280*	577**
7	0*	294*	685**	227*	206*	318*	280*	577*

The array gives the minimum costs for all of the routes. Working backwards we have:

(a) Shortest route to 2: direct from 1

(b) $1 \to 2 \to 6 \to 3$

(c) $1 \to 4$

(d) $1 \to 5$

(e) $1 \to 2 \to 6$

(f) $1 \to 4 \to 7$

(g) $1 \to 2 \to 6 \to 8$

2. Start with 26, then 56, then 45, 47, 15, 68, and finally 38. Total cost:

$24 + 187 + 111 + 53 + 206 + 259 + 108 = 948.$

3. This is a network flow problem. Add a source that directs the given supplies to the three warehouses and a sink to which the given demands from the four markets go.

If you apply the max flow on this network you derive the maximum flow to be 130. The associated minimum cut consists of $\{s, W_1, W_2, M_2, M_4\}$ and $\{W_3, M_1, M_3, n\}$ (where W_i is the ith warehouse and M_j is the jth market.

If you add capacity on any of the edges that connect the first set in the cut to the second set, then you can increase the maximum flow. For example, you can increase the capacity from W_1 to M_1 by 20 (actually 10 is enough). Here is one way to meet the demand (given the increased capacity).

	Market			
Warehouse	1	2	3	4
1	20	0	0	20
2	0	20	10	0
3	0	0	60	0

4. This is an assignment problem. Reduce the costs so that the minimum in each row is zero:

	A	B	C	D
1	15	0	0	5
2	0	50	20	30
3	35	5	0	15
4	0	65	50	70

Reduce the costs so that the minimum in each column is zero:

	A	B	C	D
1	15	0	0	0
2	0	50	20	25
3	35	5	0	10
4	0	65	45	70

Notice that there is no zero-cost assignment. So I need to reduce costs. I crossed out columns A and C and row 1 to get rid of 0s. The minimum uncrossed number is 5. I subtract this from everything, then add it back to all rows and columns with lines (to preserve non-negativity). I obtain:

	A	B	C	D
1	20	0	5	0
2	0	45	20	20
3	35	0	0	5
4	0	60	45	65

There is still no zero-cost assignment.This time I cross out rows 1 and 3 and column A. The minimum uncrossed number is 20. I subtract this from everything, then add it back to all rows and columns with lines (to preserve non-negativity). I obtain:

	A	B	C	D
1	40	0	5	0
2	0	25	0	0
3	55	0	0	5
4	0	40	25	45

Now I have a solution: 4 and A are matched, 1 and B, 2-D, 3-C or 1-D, 2-C, 3-D, and 4-A. The distance is 275.

5. If you solve the relaxed problem (the given problem without the integer constraints) you get $(x_1, x_2) = (1.5, 2.5)$ and associated value 10.5. This gives an upper bound for the given problem (with integer constraints) equal to 10. An obvious lower bound is 0, since $(x_1, x_2) = (0, 0)$ is feasible.

   Start by assigning a value to x_2. By the first constraint, $x_2 = 0, 1, 2$ or 3. If $x_2 = 3$, the problem is feasible and has the solution $(0, 3)$. Since this is in integers, the lower bound of the problem becomes 9 and this part of the tree is fathomed. If $x_2 = 2$, then the tightest constraint is $x_1 \leq 5/3$. Hence this branch of the problem can give a value no greater than 9. If $x_2 = 1$, then the solution to the problem is to set $x_1 = 2$, leading to a value of 7. Finally, when $x_2 = 0$ the highest value of the objective function is 2. Hence the maximum value of the problem is 9 and the solution is attained when $(x_1, x_2) = (0, 3)$.

6. Let the weights be 2, 2, 3 and the values be 4, 4, and 7. If $C = 4$, then the algorithm requires you to take the item first. But then you can take nothing else and you get value 7. You would do better to take the first and second items and get value 8.

7. For each of the statements below indicate whether the statement is always **TRUE**, by writing "TRUE" otherwise write "FALSE." No justification is required.

   The next three parts refer to a network in which there are N nodes and in which $c(i, j)$ is the cost of going from node i to node j and all pairs of nodes are connected. Assume that $\infty > c(i, j) \geq 0$ and that the costs are distinct ($c(i, j) = c(i', j')$ if and only if $i = i'$ and $j = j'$).

   (a) The cheapest edge (that is, the edge in which $c(i, j)$ is smallest) is always part of the minimum spanning tree.

   TRUE. The algorithm starts with this edge. Any tree without this edge could be made cheaper by including the cheapest edge.

   (b) The most expensive edge (that is, the edge in which $c(i, j)$ is the largest) is never part of the minimum spanning tree.

   TRUE. The algorithm would never select this edge. Any tree with this edge could be made cheaper by replacing it.

   (c) If the collection of the $N - 1$ cheapest edges contains no cycles, then it is a minimum spanning tree.

   TRUE. The algorithm would select these edges first as long as they create no cycles. (And $N - 1$ edges for a MST.)

   The next two parts refer to a network in which there is a source s and a sink n and in which $c(i, j) \geq 0$ is the capacity of the edge going from Node i to Node j. Denote a flow by (x_{ij}), where, for each pair of Nodes i and j, x_{ij} is the amount that flows from Node i to Node j.

   (d) There always exists a maximal flow (x_{ij}) in which either $x_{ij} = 0$ or $x_{ji} = 0$.

   TRUE. If you have a flow in which both x_{ij} and x_{ji} are positive, then you can "reverse" the flow to get (if $x_{ij} \geq x_{ji}$ for example, a new flow with $y_{ij} = x_{ij} - x_{ji}$ and $y_{ji} = 0$.

(e) Let (S, N) be a minimum capacity cut in which $s \in S$ and $n \in N$. Suppose that one forms a new network by deleting the edge connecting Node i and Node j where $i \in S$ and $j \in N$. The capacity of the maximum flow in the new network is exactly $c(i, j)$ less than the capacity of the maximum flow in the original network.

TRUE. The minimum capacity cut in the new network must be $c(i, j)$ less than the minimum capacity cut in the old network.