Econ 172A, Fall 2007: Final Examination Answers

Final median: 110, high: 146, low: 35.

Most people did well on 1, 2, 4, and 6. Some lost points on 1 and 2 for misusing the class algorithm or failing to justify an ad hoc procedure. We deducted points on the sixth question for examples that failed to answer the question. Many people did not explain their answer to the first part of Question 3. Those of you who used the max flow algorithm still needed to provide the final labeling. The typical answer to Question 5 showed little understanding of branch and bound.

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) $1 \to 2 \to 6 \to 8 \to 3$

(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 1, 1, 1, 1, 1 and the values be 1, 4, 4, 4, 9. If $C = 12$, then the algorithm requires you to take the fifth and first items (in that order) with value 10. You would do better to take the second, third, and fourth item and get value 12.

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.

   FALSE. This would be true if the network had more than two vertices, but if there is only one pair of nodes in the network, then the maximum (and minimum!) edge must be selected.

   (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 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.