November 19, 2015

Math 3012 – Applied Combinatorics Lecture 25

William T. Trotter
trotter@math.gatech.edu

Reminders

Test 3 Tuesday, November 24, 2015

Final Exam Tuesday, December 8, 2015, 8:05 – 10:55am.

Three-way Option (Full details in email)

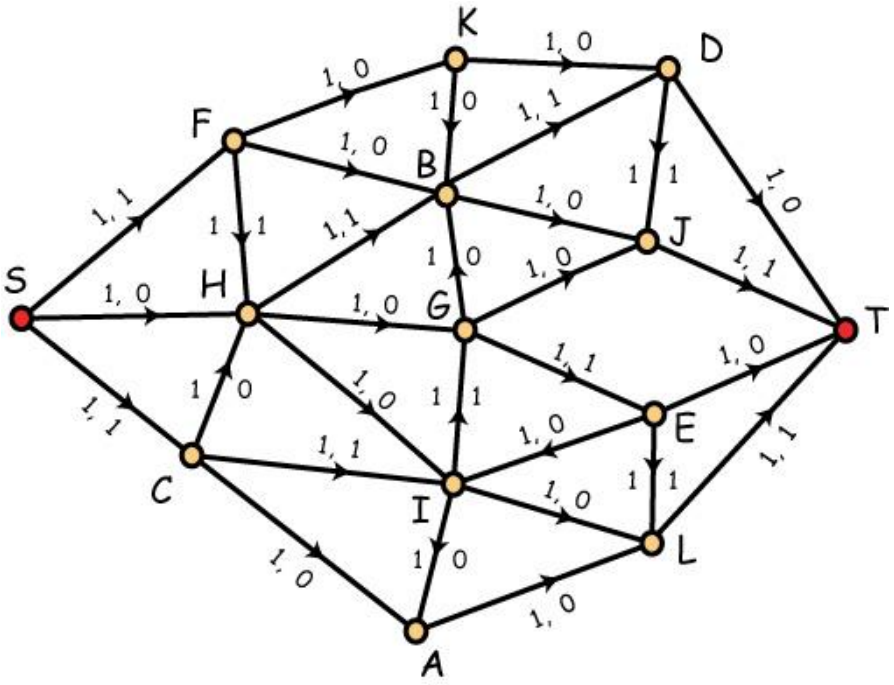
1. Do even numbered problems from assigned set.
2. Obtain/write code for implementing one of the algorithms in our course on my data set.
3. Write 3 – 4 page (typewritten) report on one of the selected math papers, all of which are accessible to undergraduates.

A Key Detail on Network Flow Problems

Fact A network flow problem posed with integer capacities on edges always has a maximum flow in which the flow on every edge is an integer. The proof of this fact is an immediate consequence of the fact that the Ford-Fulkerson labelling algorithm uses only addition, subtraction and minimum as its three operations.

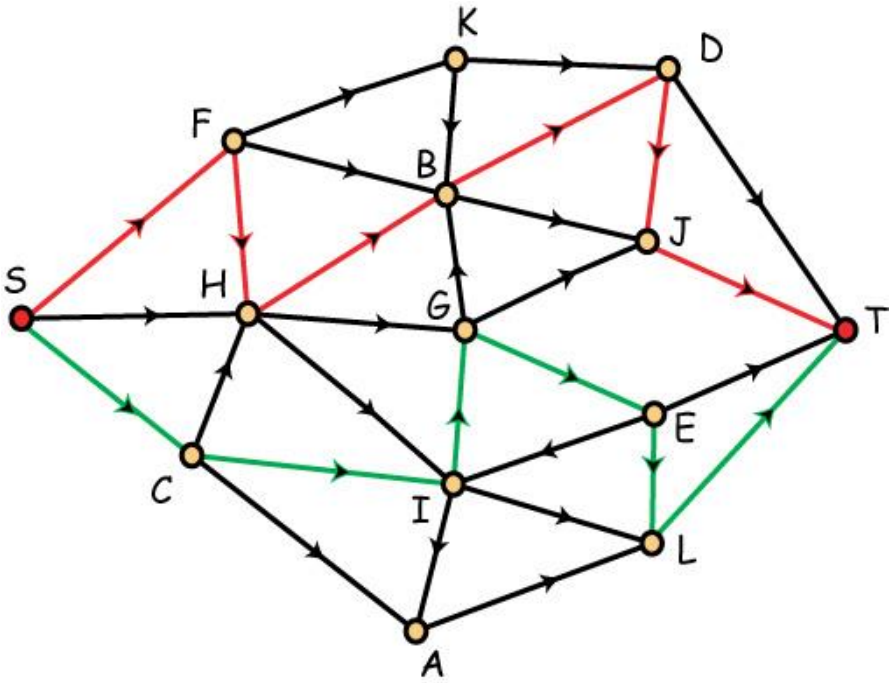
Remark It is an important general problem to determine when optimization posed with integer constraints have integer valued solutions. Network flows are just one example.

Network Flow Problems with Capacities 1



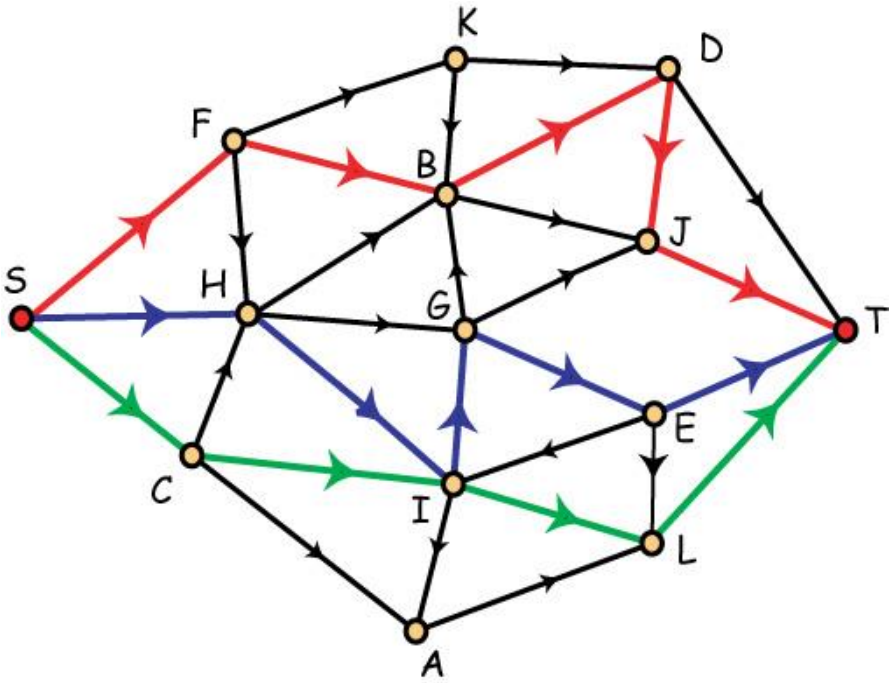
Exercise Carry out the Ford-Fulkerson labelling algorithm on the network flow.

Network Flow Problems with Capacities 1



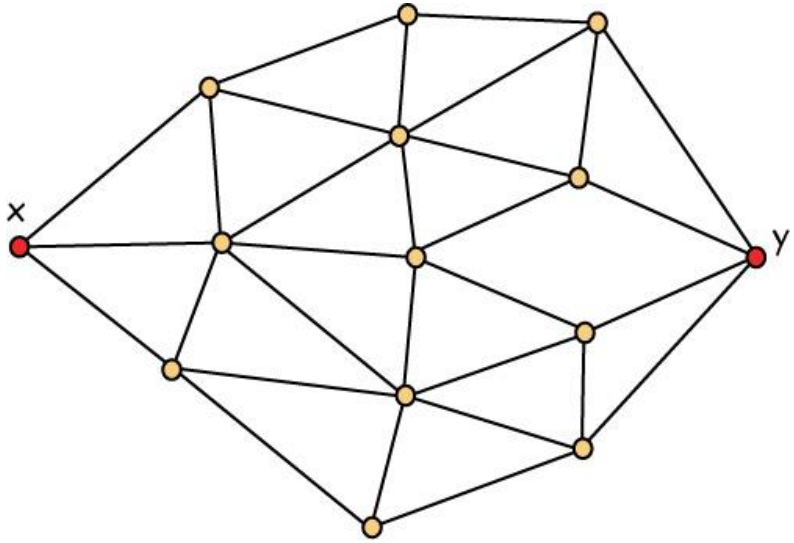
Remark Here the presence of an edge signals capacity 1, and the flow is indicated with two colors. This flow has value 2.

Network Flow Problems with Capacities 1



Remark Here the presence of an edge signals capacity 1, and the flow, which has value 3 is indicated with three colors.

Disjoint Paths From x to y



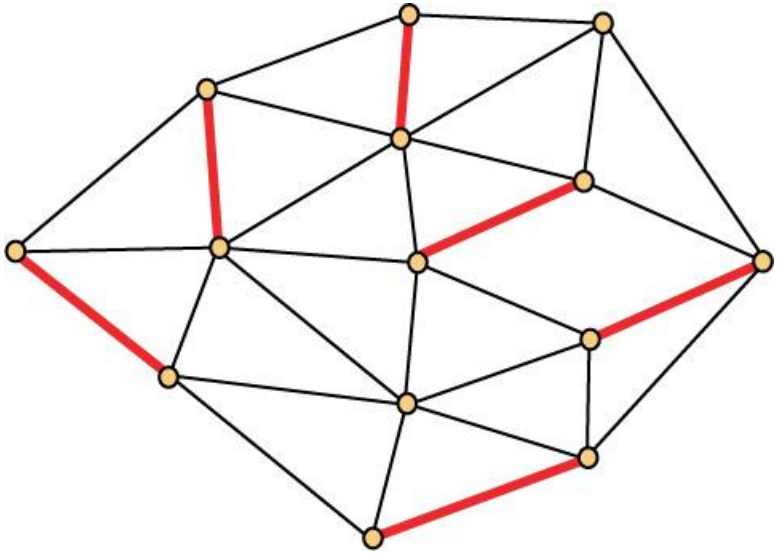
Remark There are two different notions of "disjoint". We could simply require that two different paths share no edges. Or we could make the stronger requirement that they have no vertices in common other than x and y. Network flows will find the maximum number of disjoint paths in either case.

Some Consequences

Theorem (Menger's Theorem – Edge Version) Let x and y be distinct vertices in a connected graph G. Then the maximum number of edge disjoint paths from x to y is equal to the minimum number of edges whose removal from G leaves x and y in different components.

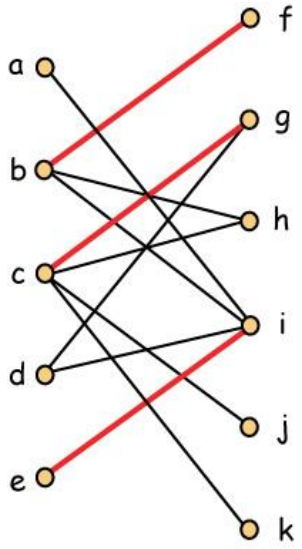
Theorem (Menger's Theorem – Vertex Version) Let x and y be distinct non-adjacent vertices in a connected graph G. Then the maximum number of vertex disjoint paths from x to y is equal to the minimum number of vertices whose removal from G leaves x and y in different components.

Matchings in Graphs



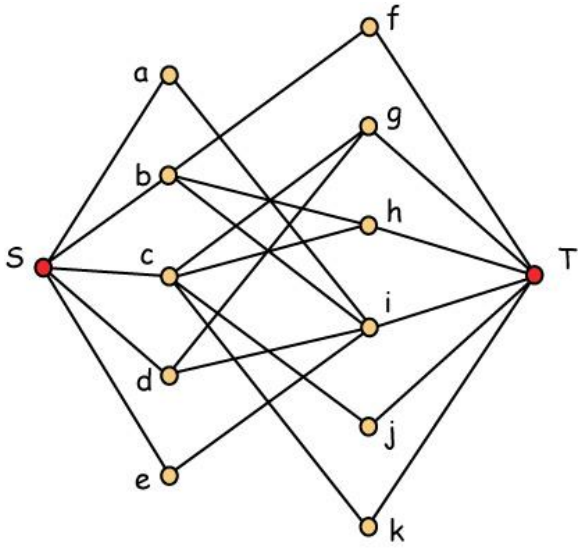
Definition A matching in a graph is a set of edges no two of which share an end point. Typically the problem is to find a maximum size matching. The matching shown is maximal. Is it maximum? The same kind of algorithm used to solve network flows will find a maximum matching in a graph.

Matchings in Bipartite Graphs



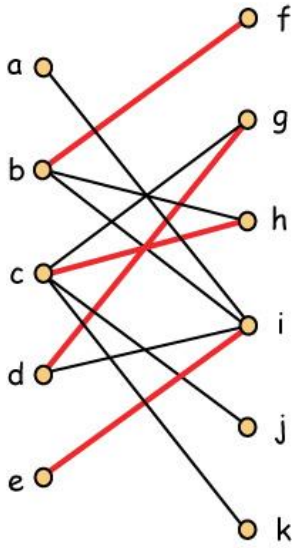
Remark We are particularly interested in finding a maximum matching in a bipartite graph. Again, the matching shown is maximal. Is it maximum?

Maximum Matchings in Bipartite Graphs



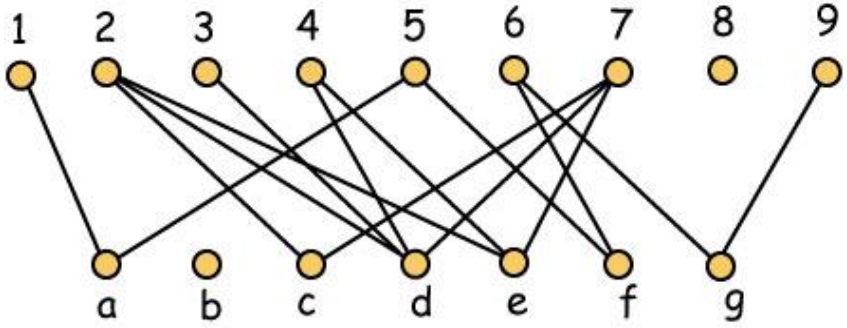
Observation There is a natural way to form a network flow problem from a bipartite graph. Simply add a source and a sink as shown, orient all edges left to right and give them capacity 1. Turn on Ford-Fulkerson and go get a cup of coffee.

Maximum Matchings in Bipartite Graphs (2)



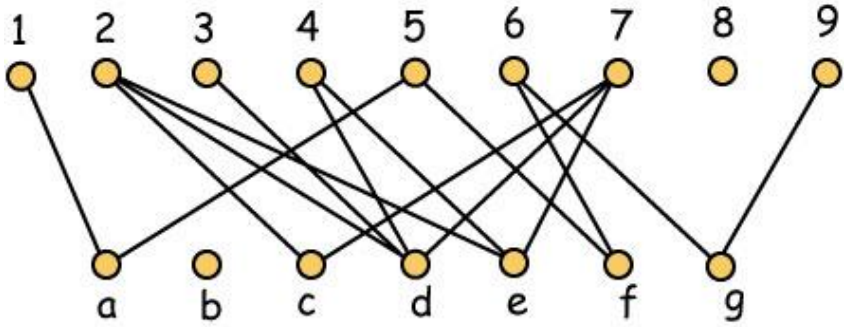
Observation It isn't really necessary to draw the source and sink as their configuration is understood. Now the matching shown is maximum. To see this, turn on Ford-Fulkerson and enjoy a donut with your coffee.

More on Matchings in Bipartite Graphs



Setup A company has 9 open positions and 7 applicants. The graph has an edge from applicant x to position i when x is capable of performing i. A matching is then an employment plan, and it is natural to try to fill as many open positions as possible. Note that some applicants may not be capable of doing any job and there may be some jobs that no applicant can do.

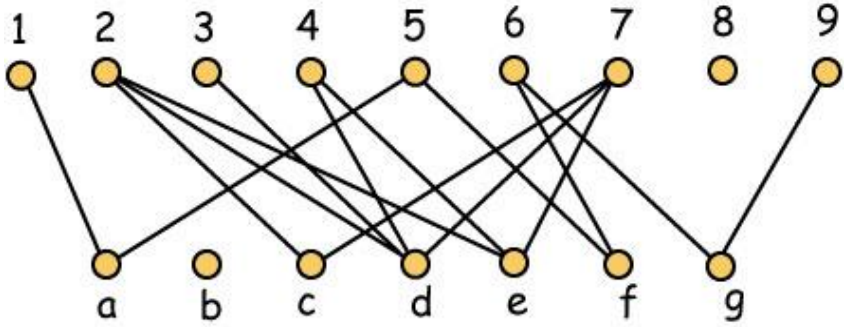
The Concept of Defect



Definitions Let $G = (X, Y, E)$ be a bipartite graph. For each subset S of X, let $N(S)$ denote the set of all elements y in Y for which there is some x in X adjacent to y. We call $N(S)$ the set of neighbors of S. The defect of G, denoted $d(G)$, is:

$$d(G) = \max \{ |S| - N(S): S \subseteq X \}$$

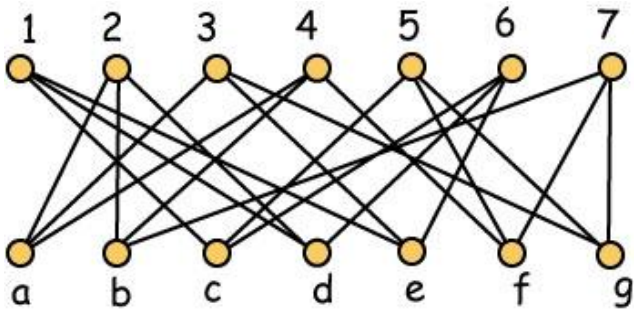
Hall's Theorem (Defect Form)

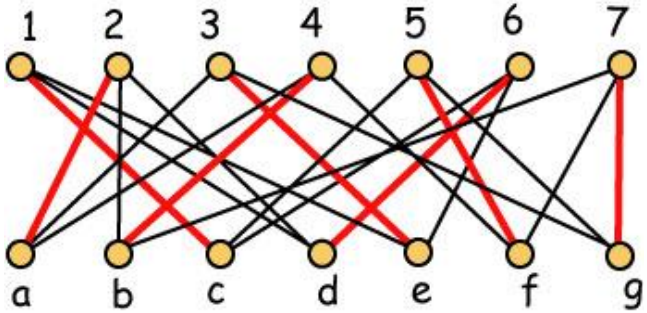


Theorem (Hall) Let $G = (X, Y, E)$ be a bipartite graph. Then the maximum size of a matching in G is $|X| - d(G)$.

Corollary There is a matching of size $|X|$ if and only if $d(G) = 0$, i.e., $|N(S)| \geq |S|$ for every subset S of X.

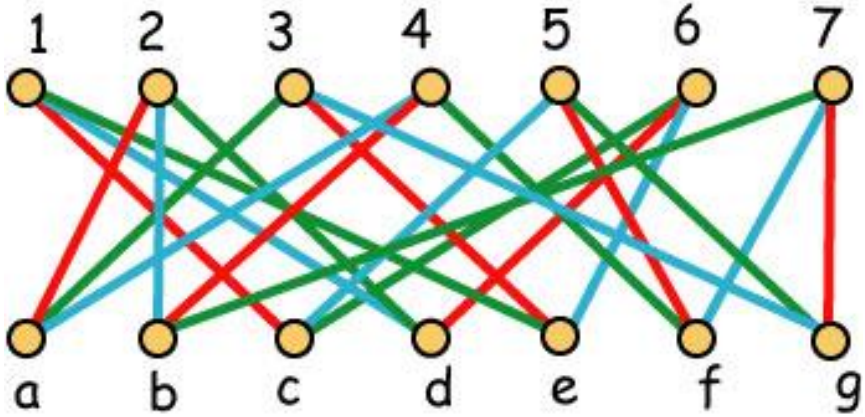
Regular Balanced Bipartite Graphs





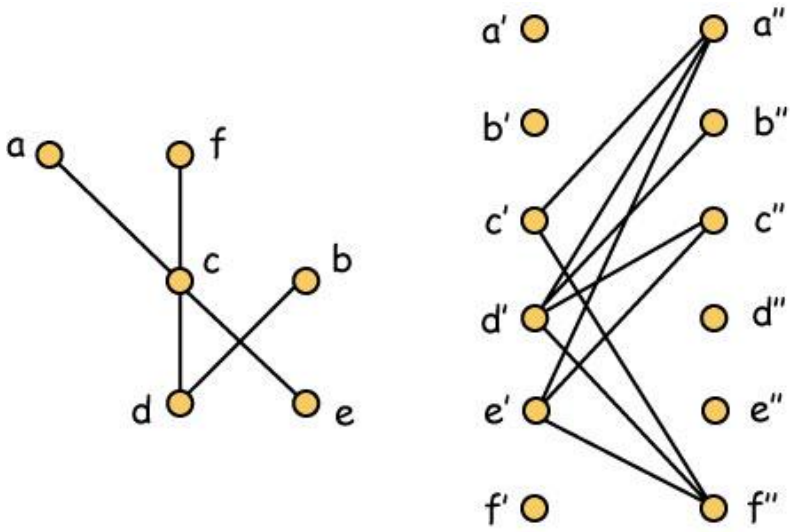
Corollary Let $G = (X, Y, E)$ be a balanced regular bipartite graph. Then there is a matching of size $|X|$ in G.

Regular Balanced Bipartite Graphs



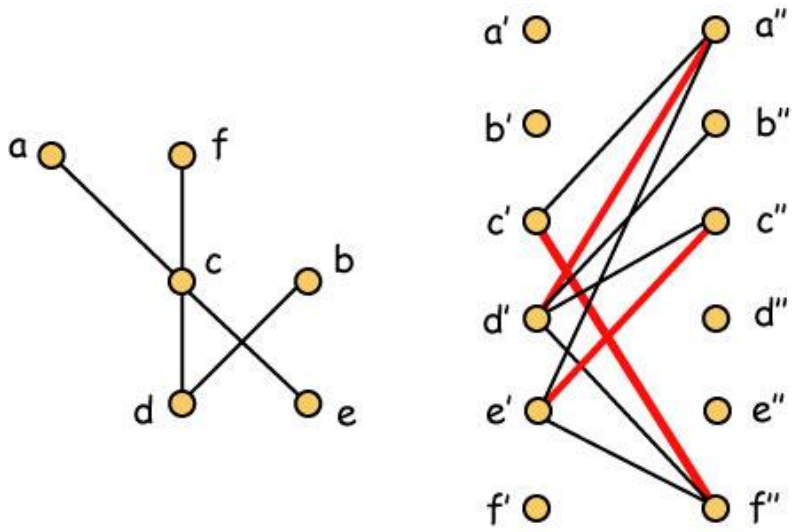
Corollary Let $G = (X, Y, E)$ be a balanced regular bipartite graph. If the degree is r, then the edge set of G can be partitioned into r matchings.

Posets to Bipartite Graphs



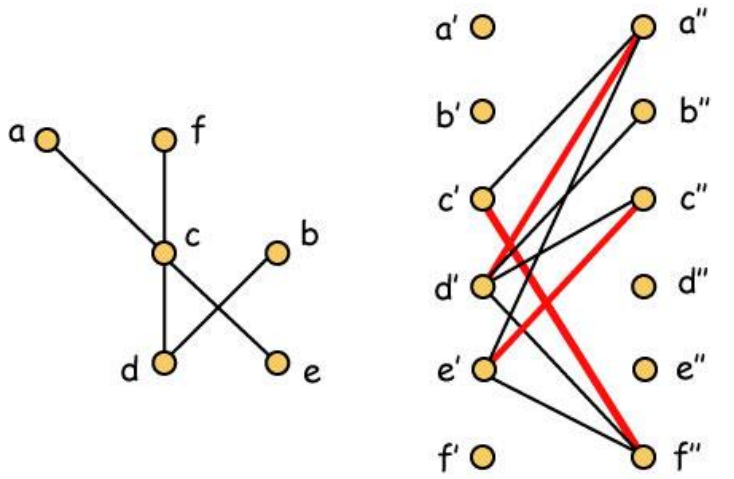
Remark For each x in P, the bipartite graph contains two points labeled x' and x'' respectively. The edges in the graph have the form $x'y''$ where $x < y$ in P.

Posets to Bipartite Graphs



Remark A matching in G determines a chain partition of P with x immediately below y in a chain when $x'y''$ is one of the matching edges. This matching corresponds to the chain partition: $C_1 = \{e < c < f\}$ $C_2 = \{d < a\}$ and $C_3 = \{b\}$. If the matching is maximum, then the chain partition is a Dilworth partition of P, i.e, it uses w chains where $w = \text{width}(P)$.

Finding a Maximum Antichain



Remark When the Ford-Fulkerson labelling algorithm halts, for each chain C_i in the partition, there is a point x_i in C_i so that x_i' is labeled and x_i'' is not. These points form an antichain. In chain $C_1 = \{e < c < f\}$, take f. In chain $C_2 = \{d < a\}$, take a and $C_3 = \{b\}$, take b (the only choice). Note that {f, a, b} is a 3-element antichain. DONE!!!