The third exam will cover the graph theory section of the class; material between Friday, October 18th and Wednesday, November 13th inclusive. The miscellaneous topics from Monday, October 14th and Friday, October 18th may also make a appearance.

The exam will not be cumulative in the sense that I won’t ask problems that are about earlier topics. But you’ll need to use many of the earlier ideas to solve problems about the new topics.

Below I try to summarize the important things we’ve covered that will be on the exam. It’s possible that I’ve missed a few topics, and it’s hard to summarize three weeks of content in two pages anyway; if you’re not sure about something, ask me.

1 Things you should know

Here are the definitions you should know:

- Totally unimodular matrices.
- Graphs, bipartite graphs, matchings, and vertex covers.
- Networks and arc capacities.
- Things you can try to find in a network: feasible s, t-flow, s, t-cut, feasible circulation, and preflow.
- The value of a flow and the maximum flow; the capacity of a cut and the minimum cut.
- The excess flow at a vertex.
- The residual graph of a flow and things related to the residual graph: residual capacity, augmenting paths.

You should know and understand the following results:

- Farkas’s lemma on the existence of solutions to $Ax \leq b$.
- That linear programs with totally unimodular constraints have integer solutions.
- The König–Egerváry theorem (I don’t think I mentioned the name. Did I?) that the size of a maximum matching in a bipartite graph equals the number of vertices in a minimum vertex cover.
• The max-flow min-cut theorem.
• Guarantees on variants of the Ford–Fulkerson algorithm and on the push–relabel algorithm.

2 Things you should be able to do

• Use Fourier–Motzkin elimination to find a feasible solution to a system of linear inequalities.
• When possible, convert a nonlinear program containing the functions $|\cdot|$, $\min\{\cdot, \cdot\}$, and $\max\{\cdot, \cdot\}$ into a linear program.
• Write down a linear program for each of the problems we talked about: bipartite matching, maximum flow, minimum cut, and so on.
  These are generally very big and so I will not be asking you to solve these linear programs.
• Determine whether a matrix is totally unimodular.
• Use Ford–Fulkerson to find a maximum flow:
  – Draw the residual graph for a network and a flow in it.
  – Use the residual graph to find augmenting paths.
  – Use an augmenting path to increase the value of a flow.
  – Use the residual graph to find a minimum cut when there is no augmenting path.
• Reduce other problems to standard maximum flow problems: circulations with demands, circulations with lower bounds, finding matchings in bipartite graphs.
• Use the push–relabel algorithm to find a maximum flow.