EECS 527 - Homework 1

Topic: Performance evaluation of the hypergraph partitioning program hMetis

The partitioning formulation
- vertex-weighted hypergraphs
  (weights correspond to gate areas -- given in benchmarks)
- no hyperedge weights
- balance constraints: each partition must contain
  42%-58% of total weight

Main action items:
- Download the hMetis program and IBM circuit benchmarks,
- Run hMetis and collect results
- Process empirical data and present as plots of Pareto curves
  (recommended plotting tools: gnuplot or Excel)

Hand in: six print-outs of plots (one per sheet);
on every sheet include your name, student ID,
the benchmark ID (e.g., ibm18)
and the number of hMetis runs
used for the first datapoint.

Download site for IBM circuit benchmarks:
http://vlsicad.cs.ucla.edu/~cheese/ispd98.html
For homework 1, you do not need to write a parser, but you will need a parser for Project 1.

Download site for hMetis:
http://www-users.cs.umn.edu/~karypis/metis/hmetis
only binaries are available
(you only need executables, but a library is also available)
You do not need to understand algorithms used by hMetis --- they will be covered later in lectures.

What to do:
- Figure out how to apply hMetis to IBM benchmarks.
  This may require converting them to the hMetis format (a converter should be available on one of the sites above)
- Figure out how to read the resulting cut (one integer for each run)
  and measure runtime (one double for each run)
- Note that every run should produce different cut because hMetis is somewhat randomized
  (you may need to change the "random seed").
  Therefore you must average cuts and runtimes over a number of runs. This will tell you what kind of outcomes you can expect "on average" from one run.
- You must figure out how many runs must be averaged so that averaging over more runs won't change the results significantly (say, by more than 2%).

- Average runtime and cut for one benchmark should be plotted in the cut-runtime plane. That will give you one datapoint.
- Additional datapoints represent cuts and runtimes that one can expect when running 2, 3 or more independent runs. In other words, instead of averages, you will now compute average-best-of-two, average best-of-three, etc.
- The curve produced by plotting those datapoints for a given benchmark is called the Pareto curve. It represents the trade-off between the solution quality and runtime.
- You must hand in Pareto curves for benchmarks ibm02, ibm04, ibm05, ibm06, ibm08 and ibm18