# Homework 3 – Due Thursday, September 15, 2016 on Canvas

Please refer to HW guidelines from HW1, course syllabus, and collaboration policy.

**Exercises** (Do not hand in.) In addition to the exercises in the lecture notes, here some favorites from KT, Chapter 4.

- (Variations on Interval Scheduling): Problems 7 ("El Goog"), 17 ("daily jobs").
- (Other exchange arguments) 12 ("bandwidth"), 13* ("photocopy service").
- (Interval perfect matching) Problem 16

**Problems to be handed in, 10 points each, 2-page limit per problem** (Don't forget to prove correctness and analyze time/space requirements of your algorithm.)

1. (**Analysis of $d$-ary heaps**) A **$d$-ary heap** is like a binary heap, described in Chapter 2.5 of Kleinberg Tardos, with the exception that non-leaf nodes have $d$ children instead of 2.
   (a) How would you represent a $d$-ary heap in an array?
   (b) Implement PARENT($i$) that, given the index $i$ of a node, returns the index of its parent and CHILD($i, k$) that, given the index $i$ of a node, returns the index of its $k$th child.
   (c) What are the minimum and the maximum number of elements in a $d$-ary heap of height $h$?
   (d) Design an efficient implementation of HEAPIFY-UP in a $d$-ary min-heap, analogous to the procedure on page 61 of KT. Analyze the running time of your algorithm in terms of $d$ and $n$.
   (e) Design an efficient implementation of HEAPIFY-DOWN in a $d$-ary min-heap, analogous to the procedure on page 63 of KT. Analyze the running time of your algorithm in terms of $d$ and $n$.
   (f) Suppose we implement a priority queue using a $d$-ary heap. Give the running times of all operations, described on pages 64–65 of KT, in terms of $d$ and $n$.
2. (**Greedy Stays Ahead**) Chapter 4, problem 5.
3. (**Greedy: A Structural Argument**) Chapter 4, problem 14.