CSCI 3100: Algorithms
Homework 4

Required Problems

1. Problem 4 from chapter 4 of the textbook: find a stabbing set of a given set of intervals.

2. Consider a long, quiet country road with houses scattered very sparsely along it. (Picture this road as a long line segment, with an eastern endpoint and a western endpoint.) Further, suppose that despite this bucolic setting, the residents of the houses are avid cell phone users. You want to place cell phone base stations at points along the road so that every house is within four miles of one of the base stations. Give an efficient algorithm that achieves this goal, using as few base stations as possible.

3. The wildly popular Spanish-language search engine El Goog needs to do a serious amount of computation every time it recompiles its index. Forturnately, the company has at its disposal a single large supercomputer, together with an essentially unlimited supply of high-end PCs.

   They've broken the overall computation into n distinct jobs, labeled $J_1, J_2, \ldots, J_n$, which can be performed completely independently of one another. Each job consists of two stages: first it needs to be preprocessed on the supercomputer, and then it needs to be finished on one of the PCs. Our notation will be that job J_i needs p_i (or $P[i]$, if you prefer to think of these as input arrays) seconds on the supercomputer, followed by f_i (or $F[i]$) seconds on a PC.

   Since there are at least n PCs available on the premises, the finishing of the jobs can be done fully in parallel, so that all the jobs can be processed on PCs at the same time. However, the supercomputer can only work on one job at a time, so the system managers need to work out an order in which to feed jobs to the supercomputer. As soon as the first job is done on the supercomputer, it will be handed to a PC for finishing; at that point in time, a second job can be started on the supercomputer. When the second job is finished on the supercomputer, it can proceed immediately to a PC for finishing regardless of whether the first job is done or not (since PCs work in parallel), and so on.

   A *schedule* is an ordering of the jobs for the supercomputer, and the *completion time* of the schedule is the earliest time at which all jobs are finished processing on PCs. This is an important quantity to minimize, since it determines how rapidly El Goog can generate a new index.

   Give a polynomial time algorithm (as fast as possible) that finds a schedule which is as small as possible.