This homework must be completed and submitted electronically. Formatting standards, submission procedures, and (optional) document templates for homeworks may be found at

http://classes.engineering.wustl.edu/cse541/ehomework/ehomework-guide.html

Advice on how to compose homeworks electronically, with links to relevant documentation for several different composition tools, may be found at

http://classes.engineering.wustl.edu/cse541/ehomework/composing-tips.html

Please remember to

- create a separate PDF file (typeset or scanned) for each problem;
- include a header with your name, WUSTL key, and the homework number at the top of each page of each solution;
- include any figures (typeset or hand-drawn) inline or as floats;
- upload and submit your PDFs to Blackboard before class time on the due date.

Always show your work.
1. The Missouri Department of Transportation has decided to plant wildflowers alongside Interstate 64 between St. Louis and Kansas City. MODOT has bid out the job and received n proposals. Say that St. Louis is at mile 0 of the interstate, while Kansas City is at mile D. Proposal $p_i$ offers to plant flowers over a (fixed) interval $[s_i, e_i)$ of the highway, $0 \leq s_i \leq e_i \leq D$, for a fee of $c_i$ dollars.

For political reasons, MODOT cannot accept only part of a proposal, so the endpoints $s_i$ and $e_i$ of each are fixed. However, it can accept two proposals that overlap, since that simply results in some parts of the highway having more flowers.

Give an efficient algorithm to choose a subset of the proposals sufficient to plant flowers along the entire route from St. Louis to Kansas City, while minimizing the total cost.

2. You are the manager for Big Ear, a special-purpose text-processing supercomputer. Research groups from political science departments around the world apply to use Big Ear to accumulate and analyze all posts to major social media sites during their local election seasons, so as to build better predictive models of who is going to win.

Each group $i$, $1 \leq i \leq n$, needs the exclusive use of Big Ear for a fixed time interval $[b_i, f_i)$, corresponding to the period leading up to a particular election. No two groups can use the machine at the same time.

The Big Ear Steering Committee has reviewed all applications to use the machine this year and has assigned each a priority score $p_i$ that estimates its scientific value. Higher (larger) priorities are better.

Give an efficient algorithm to find a non-conflicting subset of groups who will be allowed to use Big Ear this year, such that the sum of their priority scores is maximized.

3. We return to the bodies and coffins problem of the previous homework. An expedition to the mortuary’s basement has revealed an additional stash of unused coffins. Consequently, while you still need to find coffins for $n$ bodies, you now have $m \geq n$ coffins of various lengths to choose from.

Give an efficient algorithm to select $n$ of the $m$ coffins and assign them to bodies, such that the total badness of fit of your assignment (in the same sense as last time) is minimized.

4. You have a robotic vacuum cleaner in a large rectangular room with a tiled floor. There are $n \times m$ tiles; tile $(i, j)$ initially contains $s_{ij} \geq 0$ milligrams of dirt. The robot may move from a given tile to any of its four neighbors to the north, south, east, and west (but not diagonally).

The robot starts at its charging station on tile $(1, 1)$ and makes a series of moves south or east, ending up across the room at tile $(n, m)$. It then turns around and makes another series of moves north or west, ending up back at $(1, 1)$. Whenever the robot reaches a tile, it sucks up all the dirt present on the tile; subsequent visits to the same tile yield no additional dirt.

Give an efficient algorithm to compute a round trip from $(1, 1)$ to $(n, m)$ and back that maximizes the total amount of dirt sucked up by the robot.

5. You are compiling the Criterion Collection of Internet Cat Videos – a bunch of cat videos downloaded from YouTube, upconverted to 4K resolution, and stored on a series of gold-plated micro-SD cards. Due to the high price of gold, the collection must use only $k$ cards, each of which is big enough to hold $m$ bytes of video.

As input, you are given a list of $n$ videos in chronological order of their first publication, plus an ordered list of cards $c_1 \ldots c_k$. Video $i$ in the list is $\ell_i$ bytes in size. Your goal is to choose a subset of these videos to put the cards. The assignment of videos to cards must respect the input order; that
is, if videos $i$ and $j$ are both in the chosen subset, and $j > i$, then either $i$ and $j$ are on the same card, or $j$ occurs on a later card than $i$. No video may be split across cards.

Give an efficient algorithm (in $n$) to find the largest number of videos that can be placed on $k$ cards while respecting the input order. Analyze the complexity of your algorithm as a function of $n$ and $k$. (Hint: a good way to make progress is to compute the smallest number of cards needed to store (say) $s$ videos.)