1. Do Kleinberg-Tardos, Chapter 5, Problem 3 (the “bank cards” problem).

2. Suppose that $n$ people participate in a thumb-wrestling tournament; each pair of players $P_i, P_j$ have a match and the win/loss result is recorded in a matrix $M$. (So for $i < j \leq n$, we have $M[i, j] = 1$ iff $P_i$ beats $P_j$.)

A player is considered elite if they lose at most 10 matches in our tournament. Give an algorithm that runs in time $O(n \cdot \log n)$ and outputs a list of all elite players.

(Note, I’m not claiming that $n \log n$ is best-possible...)

Try, but do not turn in: Kleinberg-Tardos Chapter 6, Problems 1, 3. You may want to read over the introductory discussion of Dynamic Programming in Chapter 6, Secs. 6.1 and 6.2 (which we will also discuss in class).