Due date: Friday, April 19
Submission Instructions:

- Each group must submit one solution to this assignment in class on the due date.
- Assignments must be neatly written or typed. If the grader cannot (easily) read an answer, she will give it a 0.
- Each group must submit only one solution. If multiple solutions are handed in, the grader will randomly discard all but one of them.
- Your submission must have the name of every group member clearly marked. If your name is not on it, you do not get credit.
- All arguments should be clear, concise, and (of course) correct. You will lose points for poor writing, bad grammar, etc...
- Use of the Internet for solving these problems is strictly forbidden, and will be treated as an honor code violation.
- Inter-group work: Members of different groups may discuss the questions. However:
  - With your submission you must provide a list of all non-group members with whom any member of your group discussed the assignment.
  - No written material may leave the inter-group discussion. If you talk with someone about the assignment, you must throw away and written notes at the end of the discussion.
Problem 1 (20 points):
Suppose we looked at the currency of some randomly chosen country, and let $C$ be the sorted list of coin values. (That is, $C[0]$ is the value of the smallest count, $C[1]$ is the value of the second smallest coin, etc...) Let $k$ be the number of different coins, and assume that $C[0] = 1$.

Write an efficient algorithm that takes any value $V$ and returns the minimum number of coins that can be used to total to exactly $V$. (e.g. if it were the U.S. system and $V = 24$, the best we could do is 5).

Warning: For the U.S. currency system a greedy algorithm will work, but this is not the case for all possible currency systems.
Problem 2 (20 points): Give an $o(n^3)$ algorithm that takes a set of $n$ points on the plane and returns the largest set of co-linear points. (That is, the largest number of points that can all be connected by a single straight line.)
Problem 3 (20 points): Let $X$ and $Y$ each be a list of size $n$ numbers containing values between 1 and $n$. Create list $L$ such that $L[i] = \frac{X[i]}{Y[i]}$. Sort $L[i]$ in $O(n)$ worst-case time.