1. Implement the portfolio management appearing in the notes for Lecture 16 in any programming environment and check its performance on S&P stock data (download from http://ocobook.cs.princeton.edu/links.htm). Include your code as well as the final performance (i.e., the percentage gain achieved by your strategy).

2. Consider a set of $n$ objects (images, songs etc.) and suppose somebody has designed a distance function $d(\cdot)$ among them where $d(i, j)$ is the distance between objects $i$ and $j$. We are trying to find a geometric realization of these distances. Of course, exact realization may be impossible and we are willing to tolerate a factor 2 approximation. We want $n$ vectors $u_1, u_2, \ldots, u_n$ such that $d(i, j) \leq |u_i - u_j| \leq 2d(i, j)$ for all pairs $i, j$. Describe a polynomial-time algorithm that determines whether such $u_i$'s exist.

3. The course webpage links to a grayscale photo. Interpret it as an $n \times m$ matrix and run SVD on it. What is the value of $k$ such that a rank $k$ approximation gives a reasonable approximation (visually) to the image? What value of $k$ gives an approximation that looks high quality to your eyes? Attach both pictures and your code. (In matlab you need mat2gray function.) Extra credit: Try to explain from first principles why SVD works for image compression at all.

4. Suppose we have a set of $n$ images and for some multiset $E$ of image pairs we have been told whether they are similar (denoted $+$ edges in $E$) or dissimilar (denoted $-$ edges). These ratings were generated by different users and may not be mutually consistent (in fact the same pair may be rated as $+$ as well as $-$). We wish to partition them into clusters $S_1, S_2, S_3, \ldots$ so as to maximise:

$$\text{(number of$+\text{edges that lie within clusters}) + (\text{number of $-\text{edges that lie between clusters})}.$$ 

Show that the following SDP is an upperbound on this, where $w^+(ij)$ and $w^-(ij)$ are the number of times pair $i, j$ has been rated $+$ and $-$ respectively.

$$\max \sum_{(i,j) \in E} w^+(ij)(x_i \cdot x_j) + w^-(ij)(1 - x_i \cdot x_j)$$

$$|x_i|_2^2 = 1 \quad \forall i$$

$$x_i \cdot x_j \geq 0 \quad \forall i \neq j.$$ 

5. For the problem in the previous question describe a clustering into 4 clusters that achieves an objective value 0.75 times the SDP value. (Hint: Use Goemans-Williamson style rounding but with two random hyperplanes instead of one. You may need a quick matlab calculation just like GW.)
6. Suppose you are given $m$ halfspaces in $\mathbb{R}^n$ with rational coefficients. Describe a polynomial-time algorithm to find the largest sphere that is contained inside the polyhedron defined by these halfspaces.

7. Let $f$ be an $n$-variate convex function such that for every $x$, every eigenvalue of $\nabla^2 f(x)$ lies in $[m, M]$. Show that the optimum value of $f$ is lowerbounded by $f(x) - \frac{1}{2m} \| \nabla f(x) \|_2^2$ and upperbounded by $f(x) - \frac{1}{2M} \| \nabla f(x) \|_2^2$, where $x$ is any point. In other words, if the gradient at $x$ is small, then the value of $f$ at $x$ is near-optimal. (Hint: By the mean value theorem, $f(y) = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} (y - x)^T \nabla^2 f(z) (y - x)$, where $z$ is some point on the line segment joining $x, y$.)