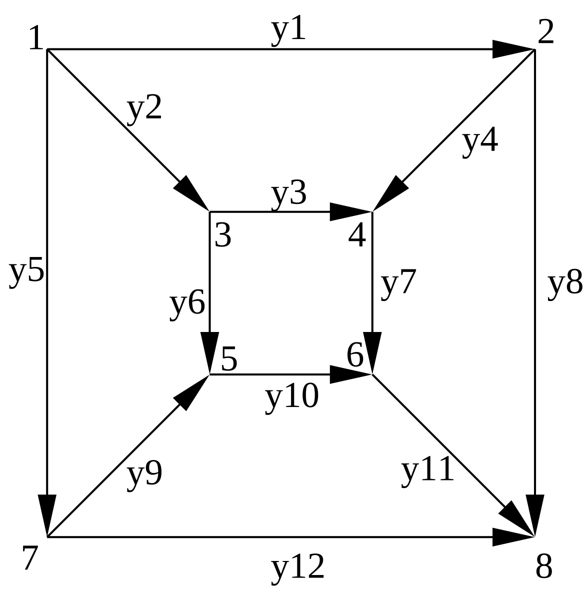
18.06, Fall 2004, Problem Set 5

Due before 4PM on Wednesday October 20th, 2004, in the boxes in 2-106. No late homework will be accepted. Don't forget to write your name, recitation section **and the names of students you have collaborated with** on the problem set. There is one box for each recitation section. For full credit, please be sure to show and explain your work. Exercises refer to the 3rd edition of the textbook.

Reading assignment: Sections 4.1, 4.2 and 8.2.

1. Consider the electrical network given below. The conductances of all edges are 1 except for the four edges 3, 6 ,7 and 10 between the nodes 3, 4, 5 and 6 for which the conductance is 2. Assume there is a source of current between nodes 1 and 8 which injects one unit into node 1 and takes 1 unit out of node 8 (this means that for the given network the net current into node 1 must be -1 (total current into node 1 minus total current out of node 1) while the net current into node 8 must be 1). Compute the resulting currents $y_1, \cdots, y_{12}$. You can use MATLAB, but you should explain what you do.



2. Show that, for any matrices A of size $m \times n$ and B of size $n \times p$, we have $r(AB) \leq r(A)$ and $r(AB) \leq r(B)$. (You can rely on exercise 3 of problem set 3.)

3. (a) Give the coordinates of n points in R^n such that the distances between any two of them are all the same. (The maximum number of such points in R^n is $n+1$.)

   (b) Here you'll show that the maximum number of points in R^n such that the distances between any pair of them are all equal is at most $n+1$ (This seems pretty obvious but one nevertheless needs to prove this fact.). Here is one way to prove this. Suppose you have p points in R^n such that all inter-distances are 1 (after scaling). You can translate the points so that one of them is at the origin. You can also assume they span the space; otherwise you can just decrease n. Now put the other $p-1$ points as the column vectors of a $n \times (p-1)$ matrix Q. Let $R = Q^T Q$.

i. What is R? (Give all entries of R.)

ii. What is the rank of R? Explain how you obtain it. (There are many ways of computing the rank; here computing $dim(N(R))$ might be easiest.)

iii. How can you now deduce that $p - 1 \leq n$? Justify.

4. In this subproblem, you will construct 50 points $w_1, w_2, \cdots, w_{50}$ in a 30-dimensional subspace of R^{50} such that the distances between any of them are all *approximately* the same using MATLAB (from the previous exercise you know that achieving precisely the same distance for all pairs of points is impossible). Here are the steps you need to do for this. Just give a printout of your MATLAB commands (the `diary` command is useful for this) and also give the answer to subquestion 4f.

(a) Generate 30 vectors $v_1, v_2, \cdots, v_{30}$ in R^{50} such that each component of each vector is independently and normally distributed. You do not need to know what "independently and normally distributed" means; you should only know that the MATLAB command $randn(m, n)$ (this is different from $rand(m, n)$) generates such an $m \times n$ matrix of independently and normally distributed entries.

(b) Compute the projection matrix P corresponding to projecting onto the subspace V spanned by the 30 vectors v_i.

(c) Take a 50×50 matrix B such that its columns correspond to 50 points in R^{50} whose distances are all the same. Now compute the projection of all these 50 points onto the subspace V using the projection matrix P above. Let W be the 50×50 matrix such that the ith column is the projection w_i of the ith column of B onto V.

(d) Compute the rank of W.

(e) Calculate the distances between w_i and w_j for any i and j. Given the matrix W, this can be done easily by using the following MATLAB commands (the matrix D will have zeroes on the diagonal and the distance between w_i and w_j as entry (i, j)).

```
> Q=W'*W;
> d=diag(Q);
> D=sqrt(d*ones(1,50)+ones(50,1)*d'-2*Q)
```

(f) Give the ratio between the maximum and the minimum distance between two vectors w_i and w_j. By repeating the steps above (say 50 times), how small a ratio can you achieve? (A convenient way to repeat things in MATLAB is to use a *for* loop; type *help for* to see how to use it.)