Prof. Friedrich Eisenbrand
Alfonso Cevallos

Discussions from: March 31, 2015

Computer Algebra

Spring 2015

Assignment Sheet 4

Note: These are just notes and not necessarily full solutions to each exercise. Please report any mistakes you may find.

Exercise 1

Recall the Gram-Schmidt orthogonalization process. It decomposes $A = BU$, where B is a matrix with pairwise orthogonal columns $b_1, \cdots, b_n$, and U is a lower-triangular matrix with ones on the diagonal. In particular, $b_1 = a_1$, and for every successor we have $a_i = b_i + \sum_{j<i} u_{i,j} b_j$. If we set $u_{i,i} = 1$, then $U = (u_{i,j})_{i,j}$ is the desired matrix above. As a consequence, we have that $\det U = 1$, so $\det A = \det B \det U = \det B$.

Furthermore, we can compute $(\det B)^2 = \det B^T B = \det(b_i^T \cdot b_j)_{i,j} = \det[|b_1|^2 e_1, \cdots, |b_n|^2 e_n] = \prod_i |b_i|^2$; and therefore $|\det A| = |\det B| = \prod_i |b_i|$. And since $a_i = b_i + \sum_{j<i} u_{i,j} b_j$, and vectors b_i are orthogonal, from Pythagoras theorem we have $|b_i| \leq |a_i|$. Finally, we get the result $|\det A| \leq \prod_i |a_i|$. Deriving from here the bound $|\det A| \leq n^{n/2} B^2$ is trivial.

Recall now Leibniz's formula for the determinant: $\det A = \sum_{\sigma \in S_n} sign(\sigma) \prod_{i=1}^n a_{i,\sigma(i)}$, where the sum ranges over all permutations σ of the set $\{1, \cdots, n\}$. As there are $n!$ such permutations, and every term in the sum is the product of n coefficients, the formula leads to the bound $|\det A| \leq n! \cdot B^n$. Now, using Stirling's approximation for the factorial: $n! \approx \sqrt{2\pi n}(n/e)^n$; for large values of n we clearly see that $n!$ is much bigger than $n^{n/2}$. Hence Hadamard bound is asymptotically better than Leibniz bound.

Exercise 2

From Hadamard bound it immediately follows that $size(\det A) = O(n \log n + n \log B)$. The important thing here is that this bound is polynomial in the input size. We will use this fact later on. Recall that Gaussian elimination iteratively performs the following three operations: swapping two rows; swapping two columns; and adding to one row a scalar multiple of another row. And at the end we obtain a matrix in row echelon form, where for each i, the first non-zero element of row i (pivot element) is strictly after the first non-zero element of row $i-1$, and where all zero rows come last.

We will not recall the exact algorithm here. But it can be easily checked that the number of basic operations is polynomial in the input size (i.e. in n and $\log B$). Hence, the only non-polynomiality problem may come from the coefficients involved becoming exponentially large (as occurs in the fast exponentiation algorith).

We can assume wlog that no swapping of rows or columns is ever performed, as these operations do not change the size of the coefficients. Then, throughout the algorithm, to each row i we only add a linear combination of the rows above: $1, \cdots, i-1$. This implies that the determinant of each submatrix of A containing all the first i rows stays the same throughout the algorithm. Now, for $i = 1, \cdots, rank(A)$, let $(i, g(i))$ be the pivot of row i. Consider any element $a_{i,j}$ after we reduced in row echelon form the first k rows of the matrix (k-th step). We want to establish that, for each i, j, k, the coefficient $a_{i,j}$ is of size polynomial in the input size. The only interesting case is when $k \leq i$ and, if $k = i$, when $j \geq g(i)$ (during steps $k > i$, row i is not modified; and for $k = i$ and $j < g(i)$, $a_{i,j} = 0$).

Let A' be the matrix A after the k-th step, and let B, C be its two submatrices defined as follows: B has rows $1, \cdots, \min\{k, i-1\}$, and columns $g(1), \cdots, g(\min\{k, i-1\})$. C is obtained from B by adding row i and column j. Clearly, both matrices are upper triangular, so their determinant is the product of the diagonal entries. Hence we obtain $a_{i,j} = \frac{\det C}{\det B}$. We deduce that the size of $a_{i,j}$ is at most the size of $\det C$, for which we can use the same bound we got for $\det A$, which is polynomially bounded. This completes the proof.

Exercise 3

Modulo 3: $A = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 2 & 1 \\ 0 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 0 \end{pmatrix}$, so $\det A = 1 * 2 * 0 = 0$.

Modulo 5: $A = \begin{pmatrix} 1 & 0 & 3 \\ 2 & 4 & 1 \\ 0 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{pmatrix}$, so $\det A = 1 * 4 * 2 = 3$.

Modulo 7: $A = \begin{pmatrix} 1 & 0 & 5 \\ 2 & 6 & 1 \\ 0 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 5 \\ 0 & 6 & 5 \\ 0 & 0 & 5 \end{pmatrix}$, so $\det A = 1 * 6 * 5 = 2$.

Therefore, by use of the Chinese remainder theorem, we are able to say that $\det A \equiv 93 \mod 105$ (while only working with single-digit numbers!). So $\det A = 93 + 105k$, for some integer k; but what is the precise value? In matrix A, all the coefficients are bounded in absolute value by $B = 2$. Therefore, Leibniz bound gives us $|\det A| \leq B^n n! = 2^3 3! = 48$; which implies that $-48 \leq \det A \leq 48$ (a range of $2 * 48 + 1 = 97$ possible values). Since $105 > 97$, there is only one possible value of $\det A = 93 + 105k$ within this range, and this is $\det A = -12$. .

Exercise 4

An algorith for solving the problem is the following: a) Transform all entries of A and b into integers. b) Apply Gaussian elimination in order to reduce the augmented matrix $(A|b)$ into a matrix $(A'|b')$ in row echelon form, and recall that the set of feasible solutions to $Ax = b$ coincide with the set of feasible solutions to $A'x = b'$. c) Apply the Gauss-Jordan elimination in order to reduce $(A'|b')$ into a matrix $(A"|b")$ in reduced row echelon form. Again, the set of feasible solutions to $Ax = b$ coincides with the set of feasible solutions to $A"x = b"$, and a solution of the latter can be immediately found.

One easily checks that all these steps require a number of basic operations that is polynomial in n and m, so the only non-polynomiality problem could come from an exponential increase in the size of the numbers. From exercise 2, we know this cannot happen in the second step, and the third step is also safe, by a similar proof.[1] Here, we show that the size of the numbers stays polynomial after the first step.

Note that it suffices to multiply all numbers by the least common multiple N of all denominators of the entries of A and b. We have $size(N) \leq size(\prod_{i,j} a_{i,j} \prod_i b_i) = O(\sum_{i,j} size(a_{i,j}) + \sum_i size(b_i)) = O(size(A) + size(B))$. Hence we can assume withouth loss of generality that all the inputs are integral.

Exercise 5

Part 1. Let G be an n-vertex graph, and $T(G)$ its Tutte matrix. Let M be a matching of G of maximum cardinality (so $\nu(G) = |M|$), and let H be the subgraph of G induced by its endpoints. So clearly M is a perfect matching in H, and $2\nu(G) = 2|M| = 2\nu(H) = rank(T(H))$, by Tutte's theorem. We also have $rank(T(H)) \leq rank(T(G))$, because the Tutte matrix of H is a submatrix of the Tutte matrix of G (obtained by keeping only the rows and columns indexed by $V(H)$). The following claim completes the proof: $rank(T(G)) \leq 2\nu(G)$.

Suppose it is not the case, so $rank(T(G)) = r$, for some $r > 2\nu(G)$. The row-space of $T(G)$ has a basis of r rows, indexed by a set of vertices $X \subset V(G)$. Let T' be the $r \times n$ submatrix of $T(G)$ obtained by keeping only these rows. As it has full row-rank, its column-rank must be r. Now, consider a vertex $v \in V(G) \setminus X$: the corresponding row is a linear combination of the rows of X, and since $T(G)$ is skew-symmetric, the corresponding column in $T(G)$ is also a linear combination of the columns of X. The removal of some coordinates won't change this vector dependence, so also in T' we will have that the column of v is a linear combination of the columns of X. Therefore, the columns of X generate the column-space of T', and their number matches the column-rank, so they must also be a basis of the column-space. Finally, if T'' is the $r \times r$ submatrix of T' obtained by keeping only these columns, it is a full-rank square matrix. And it clearly corresponds to the Tutte matrix of the subgraph of G induced by X. By Tutte's theorem, this subgraph must have a perfect matching, of size $r/2 > \nu(G)$, which is also a matching of G, and this leads to a contradiction.

Part 2. Fix a $2n$-element set S (for instance $S = \{1, \cdots, 2n\}$) and let an *evaluation* of the Tutte matrix $T(G)$ be the matrix $\bar{T}$ obtained by substituting to the indeterminates of $T(G)$ numbers chosen uniformly at random from S. The algorithm is the following: generate k evaluations of $T(G)$; compute their ranks; and output the maximum of the ranks, divided by 2.

[1] We refer the reader to *Geometric Algorithms and Combinatorial Optimization* by Grötschel, Lovasz and Schrijver, Springer-Verlag, New York, 1988. Pages 36-39. Book available here: http://www.zib.de/groetschel/pubnew/paper/groetschellovaszschrijver1988.pdf

We claim that the algorithm outputs the correct value of $\nu(G)$ with probability at least $1 - 2^{-k}$; and to prove this it is enough to prove that one iteration will find the right answer with probability at least $1/2$. Let H be a subgraph of G such that $2\nu(G) = 2\nu(H) = rank(T(H)) = rank(T(G))$ (as in part 1.), and keep in mind that $T(H)$ is a submatrix of $T(G)$, of full rank. Let $\bar{T}$ be an evaluation of $T(G)$. It is clear that $rank(\bar{T}) \leq rank(T(G))$, and we just need to bound the probability of an inequality. If $\bar{T}_H$ is the submatrix of $\bar{T}$ corresponding to $T(H)$, then: $P[rank(\bar{T}) \leq rank(T(G))] \leq P[rank(\bar{T}_H) \leq rank(T(H))] = P[\det(\bar{T}_H) = 0 \mid \det(T(H)) \neq 0] \leq 1 - \frac{n}{|S|} = \frac{1}{2}$. The last inequality follows from the Schwartz-Zippel lemma.

Exercise 6
See the code.