determinants and permanents

permutations

A **permutation** is a 1-1 map of a set X onto itself. The number of permutations on an n-element set is $n!$. A simple inductive proof shows that, for all $n \geq 4$, $2^n \leq n! \leq 2^{n^2}$. That is: $n \mapsto n!$ is $\Omega(2^n)$ and $O(2^{n^2})$.

transpositions

A **transposition** is a permutation of two elements, i.e $\sigma = (\alpha\beta)$.

parity

The **parity** of a permutation σ, denoted $sgn(\sigma)$ is 1 if σ is the product of an even number of transpositions, -1 otherwise.

proof that $sgn(\sigma) = \pm 1$

Let $\sigma \in S_n$. Write $\sigma = (\alpha_1\alpha_2\cdots\alpha_m)$. In the view of

$$(\beta_1\beta_2\beta_3\cdots\beta_k) = (\beta_1\beta_2)(\beta_1\beta_3)\cdots(\beta_1\beta_k)$$

any permutation of length k can be written as a composite of $k-1$ transpositions. Now consider when k is even or odd, then the result follows.

Note that $sgn(\sigma \cdot \tau) = sgn(\sigma) \cdot sgn(\tau)$ and $sgn(t) = -1$, where t is a transposition.

matrices

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}$$

$$det(A) = \sum_{\sigma \in \text{perm}\{1,\cdots,n\}} sgn(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

$$permanent(A) = \sum_{\sigma \in \text{perm}\{1,\cdots,n\}} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

calculating the determinant

We can convert the matrix into upper triangular form, then we know the determinant is the product of the elements across the diagonal. To zero the i column below the diagonal, we need to do a transposition of columns $O(n)$ and for each of $(n-i+1) \leq n$ rows below the ith row, subtract a multiple of the ith row from that row, which contains $n-i+1 \leq n$ non-zero elements, so $O(n)$ operations per row. So cost of zeroing the ith column below the diagonal is $O(n^2)$. There are $n-1 \leq n$ columns to zero below the diagonal, so cost of converting to UT form is $O(n^3)$ and cost of multiplying diagonal elements is $O(n)$. Hence total cost is $O(n^3+n) = \boldsymbol{O(n^3)}$.

calculating the permanent

There are no efficient methods to calculate the permanent - the best-known algorithms run in exponential time.

lexicographic order

Consider a finite set A which is totally ordered. Given two different elements of the same length $\alpha_1\alpha_2\cdots\alpha_k$ and $\beta_1\beta_2\cdots\beta_k$, the first sequence is smaller than the second one for lexicographic order, if $a_i < b_i$ for the first i where a_i and b_i are different.

If one sequence is shorter than another, then pad it with "blank" characters - a character than is treated as smaller than every element of A.

$\boldsymbol{\Omega(n \log n)}$ for comparison-based algorithm

Suppose we want to sort n elements. There are $n!$ permutations of these n elements. If we draw a binary tree with each leaf represents a permutation of these n elements, the number of comparisons we need at most is the height of tree. A tree with height h has at most 2^h leaves, then we have $n! \leq 2^h$, it follows that $log(n!) \leq h$. In the view of

$$n! > (\frac{n}{2})^{\frac{n}{2}} \text{ for } n \geq 1$$

we know that $h \geq log(n!) \geq log(\frac{n}{2})^{\frac{n}{2}} = (\frac{n}{2})\, log\, \frac{n}{2}$ so it follows that $h \in \Omega(n \log n)$.

notes