Part I. Linear Algebra (50%)

1. Let B be an ordered basis $\left\{\begin{bmatrix}1\\1\end{bmatrix},\begin{bmatrix}-1\\100\end{bmatrix}\right\}$ for R^2 and $P=\begin{bmatrix}1/\sqrt{2} & -1/\sqrt{2}\\ 1/\sqrt{2} & 1/\sqrt{2}\end{bmatrix}$.

   In a series of alternations between a change of basis and a linear transformation in R^2, all the changes of basis are based on the same transition matrix P and all the linear transformations share the same operation L. The series begins with a change of basis represented by the transition matrix P from the basis B to a new basis B_1, and subsequently a linear transformation $[L(\mathbf{v})]_{B_1}=P[\mathbf{v}]_{B_1}$ is applied to R^2 with respect to the ordered basis B_1. Next, the bases are changed from B_1 to B_2 using the same transition matrix P, followed by the linear transformation $[L(L(\mathbf{v}))]_{B_2}=P[L(\mathbf{v})]_{B_2}$ with respect to the ordered basis B_2. Repeating the above step for n times (n>0) gives $[L^n(\mathbf{v})]_{B_n}=P[L^{n-1}(\mathbf{v})]_{B_n}$ with respect to a new basis B_n. (Note: $L^n(\mathbf{v})=L(\cdots L(L(\mathbf{v})))$ denotes n operations on $\mathbf{v}$.)

   (a) What is the ordered basis B_2? (10%)

   (b) What is the minimum value of n that satisfies $[L^n(\mathbf{v})]_{B_n}=[\mathbf{v}]_B$? (10%)

2. Let A be an n x n real symmetric matrix and E be a matrix whose columns are the eigenvectors of A corresponding to the eigenvalue 1 of multiplicity n. If L is a linear transformation $L(\mathbf{v})=E^T\mathbf{v}$ for any v in R^n, show that $\|L(\mathbf{v})\|^2=trace(A^{-1}\mathbf{v}\mathbf{v}^T)$. (15%)

3. Find elementary matrices $E_1, \ldots, E_K$ such that $A=BE_1\ldots E_K$, where

$$A=\begin{bmatrix}0&3&1\\0&1&0\\4&0&0\\4&0&0\end{bmatrix} \text{ and } B=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\0&0&1\end{bmatrix}. \text{(15\%)}$$

(背面仍有題目,請繼續作答)

Discrete Mathematics 2005 (50%)

1. *Divide-and-conquer* algorithm is used to solve a problem by the following two steps.
   Step 1: Breaking the given problem of size $\boldsymbol{n}$ into $\boldsymbol{a}$ smaller problems of the same type and the same size, either $\lceil n/b \rceil$ or $\lfloor n/b \rfloor$, where $\boldsymbol{a}, \boldsymbol{b} \in \mathbf{Z}^{+}$.
   Step 2: Solving the $\boldsymbol{a}$ smaller problems and use their solutions to construct a solution for the original problem of size $\boldsymbol{n}$.
   (a) [5%] Find the recurrence relation of the time complexity function $\boldsymbol{f(n)}$ of the *divide-and-conquer* algorithm. If the time to solve the initial problem of size $\boldsymbol{n} = \mathbf{1}$ is a constant $\boldsymbol{c} \geq \mathbf{0}$, and the time to break the given problem into smaller problems, together with the time to combine the solutions of these smaller problems to get a solution for the given problem, is $\boldsymbol{h(n)}$.
   (b) [20%] For $\boldsymbol{n} = \boldsymbol{b}^{k}$, prove that
   (1) $f(n) = c(\log_b n + 1)$, when $\boldsymbol{a} = \mathbf{1}$, and (2) $f(n) = \dfrac{c(an^{\log_b a} - 1)}{a-1}$, when $\boldsymbol{a} \geq \mathbf{2}$.
   (c) [5%] Apply the above results to determining the worst-case time complexity for the *binary-search* algorithm.

2. [10%] Find a generating function for the number of ways to partition a positive integer $\boldsymbol{n}$ into positive-integer summands, where each summand appears an odd number of times or not at all.

3. [10%]
   (a) Construct a state diagram for a finite state machine that recognizes any input string with an even number of **1**s.
   (b) Construct a state diagram for a finite state machine that recognizes any input string that contains at least one **1** and at least one **0**.