

Chương 7: Kỹ thuật Biến đổi để trị (Transform-and-Conquer)

Phép khử Gauss

Giải thuật

```
GaussElimination(A[1 .. n, 1 .. n], b[1 .. n]) {
    for (i = 1; i <= n; i++)
        A[i, n + 1] = b[i];
    for (i = 1; i < n; i++)
        for (j = i + 1; j <= n; j++)
            for (k = i; k <= n + 1; k++)
                A[j, k] -= A[i, k] * (A[j, i] / A[i, i]);
}
```

Giải thuật (cải tiến)

```
BetterGaussElimination(A[1 .. n, 1 .. n], b[1 .. n]) {
    for (i = 1; i <= n; i++)
        A[i, n + 1] = b[i];
    for (i = 1; i < n; i++) {
        pivotrow = i;
        for (j = i + 1; j <= n; j++)
            if (|A[j, i]| > |A[pivotrow, i]|)
                pivotrow = j;
        for (k = i; k <= n + 1; k++)
            swap(A[i, k], A[pivotrow, k]);

        for (j = i + 1; j <= n; j++) {
            temp = A[j, i] / A[i, i];
            for (k = i; k <= n + 1; k++)
                A[j, k] -= A[i, k] * temp;
        }
    }
}
```

Phân rã LU (LU decomposition)***Giải thuật***

```

LU_Decomposition(A[1 .. n, 1 .. n]) {
    for (i = 1; i < n; i++) {
        pivotrow = i;
        for (j = i + 1; j <= n; j++)
            if (|A[j, i]| > |A[pivotrow, i]|)
                pivotrow = j;
        for (k = 1; k <= n; k++)
            swap(A[i, k], A[pivotrow, k]);

        for (j = i + 1; j <= n; j++) {
            temp = A[j, i] / A[i, i];
            for (k = i + 1; k <= n; k++)
                A[j, k] -= A[i, k] * temp;
            A[j, i] = temp;
        }
    }
}

```

Luật Horner***Giải thuật***

```

Horner(P[0 .. n], x) {
    p = P[n];
    for (i = n - 1; i ≥ 0; i--)
        p = x * p + P[i];
    return p;
}

```

Lũy thừa nhị phân (Binary Exponentiation)***Giải thuật***

```

LeftRightBinExp(a, b[0 .. M]) {
    product = a;
    for (i = M - 1; i ≥ 0; i--) {
        product *= product;
        if (b[i] == 1)
            product *= a;
    }
    return product;
}

```

Giải thuật

```
RightLeftBinExp(a, b[0 .. M]) {  
    term = a;  
    (b[0] == 1) ? product = a : product = 1;  
    for (i = 1; i ≤ M; i++) {  
        term *= term;  
        if (b[i] == 1)  
            product *= term;  
    }  
    return    product;  
}
```