

Universidad Nacional Autónoma de México

Facultad de ingeniería

Estructura de datos y algoritmos I

Problemas NP y P

Hurtado Rodriguez Nestor Rafael

Grupo:13

El problema $P=P_n$ es uno de los siete problemas del milenio, dicho problema ha puesto a trabajar del área de la computación y de las matemática. Estudio como crece el costo computacional cuando se le introduce determinado problema en relación al crecimiento de dicho problema.

En la teoría de la complejidad se dice que el problema $P=P_n$ es un conjunto de problemas en los que podemos comprobar en un tiempo razonable (Polinomial) si una respuesta al problema es correcta o no. Se le llama P al conjunto de problemas en los que podemos encontrar una respuesta al conjunto de problema en un tiempo razonable. Los problemas P_n contienen a todos los problemas, pero también contiene otros problemas más.

El ejemplo más común de un problema P es el de la multiplicación de dos matrices

```
#include <stdio.h>

void main(void)
{
    float mat1[3][2], mat2[2][3], resultado=0, resul[3][3];
    int i, j, c;
    for(i=0; i<3; i++)
        for(j=0; j<3; j++)
            resul[i][j]=0;
    for(i=0; i<3; i++){
        for(j=0; j<2; j++){
            printf("\nintroduzca un valor mat1[%d][%d]: ", i, j);
            scanf("%f", &mat1[i][j]);
        }
    }
    for(i=0; i<2; i++){
        for(j=0; j<3; j++){
            printf("\nintroduzca un valor mat2[%d][%d]: ", i, j);
            scanf("%f", &mat2[i][j]);
        }
    }
    for(i=0; i<3; i++){
        for(c=0; c<3; c++){
            for(j=0; j<2; j++){
                resultado=resultado+(mat1[i][j]*mat2[j][c]);
            }
            resul[i][c]=resultado;
            resultado=0;
        }
    }
    /*Matriz 1*/
    printf("\nMatriz 1\n");
    for(i=0; i<3; i++){
        for(j=0; j<2; j++){
            printf("\t%.2f", mat1[i][j]);
        }
        printf("\n");
    }
    /*Matriz 2*/
    printf("\nMatriz 2\n");
    for(i=0; i<2; i++){
        for(j=0; j<3; j++){
            printf("\t%.2f", mat2[i][j]);
        }
        printf("\n");
    }
    /*Matriz multiplicación*/
    printf("\nMatriz resultado\n");
    for(i=0; i<3; i++){
        for(c=0; c<3; c++){
            printf("\t%.2f", resul[i][c]);
        }
        printf("\n");
    }
}
```

Para esta tenemos que se resuelve en un tiempo polinomial $O(n^2)$

```

#include <conio.h>
#include <stdio.h>

int main()
{
    int n1, n2, producto, suma;

    printf( "\n  Introduzca primer numero (entero): " );
    scanf( "%d", &n1 );
    printf( "\n  Introduzca segundo numero (entero): " );
    scanf( "%d", &n2 );

    suma = n1 + n2;
    producto = n1 * n2;

    printf( "\n  La suma es: %d", suma );
    printf( "\n\n  La multiplicaci%cn es: %d", 162, producto );

    getch(); /* Pausa */

    return 0;
}

```

otro ejemplo es el de la suma de dos números, y le corresponde 2n

El ejemplo para los problemas Pn es un algoritmo que eleva a una potencia un número

```

#include <stdio.h>
#include <math.h>

int main(int argc, char const *argv[])
{
    double numero; // Aquí vamos a almacenar lo que el usuario introduzca
    printf("Escribe un numero y te dire su raiz cuadrada\n");
}

```

```
scanf("%lf", &numero); // Escanear lo que el usuario ingresa
double raizCuadrada = sqrt(numero); // Calcular raíz
printf("La raíz cuadrada de %lf es %lf\n", numero, raizCuadrada);
return 0;
}
```

El Instituto Clay de Matemáticas (Cambridge, Massachusetts) premia con un millón de dólares a quién sea capaz de lograr la resolución de esta conjetura.

Es un gran problema que muchos científicos computacionales y matemáticos han intentado resolver, si resulta que $P=P_n$ muchas cosas que ya se daban por sentadas cambiarán drásticamente, pero si $P \neq P_n$ entonces poco será lo que cambie.

Referencias

https://elpais.com/tecnologia/2017/05/19/actualidad/1495202801_698394.html

<https://www.xatakaciencia.com/matematicas/p-versus-np-nunca-lo-entendiste>

<https://www.youtube.com/watch?v=UR2oDYZ-Sao>