Instituto Politécnico Nacional

Escuela Superior de Cómputo

Estructuras de Datos

Práctica no. 5:

El problema de las N-Reinas

Profesor: Edgardo Adrián Franco Martínez

Integrantes:

* Calva Hernández José Manuel
* González Núñez Daniel Adrián
* Ruíz López Luis Carlos

Grupo: 1CM7

**Introducción**

La resolución de algunos problemas exige probar sistemáticamente todas las posibilidades que

pueden existir para encontrar una solución. Los algoritmos de *backtracking* utilizan la recursividad

para probar cada una de las posibilidades de encontrar la solución.

Dos son las características principales de los algoritmos de *backtracking*:

1. *Búsqueda exhaustiva:* Con todas las posibilidades o alternativas, de soluciones (tareas) parciales que conducen a la solución del problema.
2. *Backtracking*: Si una solución parcial no conduce a la solución global del problema, se vuelve atrás (se hace *backtrack*) para probar otra solución parcial.

Este método de resolución de problemas recurre a realizar una búsqueda exhaustiva, sistemática, de una posible solución al problema planteado. Descomponen el proceso de búsqueda o tanteo en tareas parciales. Cada *tarea parcial* realiza las mismas acciones que la tarea anterior y por eso se expresa, frecuentemente, de forma recursiva.

Por ejemplo, suponga el siguiente problema: determinar los sucesivos saltos que debe hacer un caballo, desde una posición inicial cualquiera, (con su forma típica de moverse) para que pase por todas las casillas de un tablero de ajedrez vacío. La *tarea parcial* consiste en realizar un salto válido, en cuanto a que esté dentro de las coordenadas del tablero y que no haya pasado ya por la casilla destino. En este problema todas las posibilidades son los ocho saltos que puede realizar un caballo desde una casilla dada.

El proceso general de los algoritmos de *vuelta atrás* se contempla como un método de prueba o búsqueda, que gradualmente construye tareas básicas y las inspecciona para determinar si conducen a la solución del problema. Si una tarea no conduce a la solución, se prueba con otra tarea básica. Es una prueba sistemática hasta llegar a la solución, o bien determinar que no la hay al haberse agotado todas las posibilidades de crear tareas.

**Planteamiento del problema**

Procedimiento *Reinas* Declarar n, a←0, b←0 Como Entero  
 Leer n // Tamaño tablero  
 Si a<n y b<n Entonces  
 Para i←1 Hasta n Con paso +1  
   
 Fin Para  
 Para i←0 Hasta n Con paso +1  
   
 Fin Para  
 Fin Si  
Fin Procedimiento

**Diseño y funcionamiento de la solución** (Descripción de la abstracción del problema y su solución, apoyándose de diagramas y figuras en un lenguaje claro) \*Descripción del funcionamiento de los algoritmos que se apoyan del TAD pila.

**Implementación de la solución** (Según la solución diseñada como se implemento en el lenguaje de programación)

**Funcionamiento** (Verificación de la solución, pruebas y resultados de salida \*Pantallazos)

**Errores detectados** (Si existe algún error detectado, el cuál no fue posible resolver o se desconoce el motivo y solo ocurre con ciertas condiciones es necesario describirlo)

**Posibles mejoras** (Describir posibles disminuciones de código en la implementación o otras posibles soluciones)

**Conclusiones** (Por cada integrante del equipo)

**Anexo (Códigos fuente \*con colores e instrucciones de compilación)**