

Tarea D: Las N Reinas y Montecarlo

Jueves 24 de Octubre

I. DESCRIPCIÓN GENERAL

En esta tarea, se desplegará una tabla con las cantidades de nodos que 3 algoritmos para resolver el problema de las N reinas considerarían. También se mostrará el estimado que una simulación de Montecarlo encuentra. Toda la programación debe hacerse en Lenguaje C sobre Linux. La salida de esta tarea es a consola.

II. N REINAS

Tenemos 3 algoritmos para resolver el problema de las N Reinas: exhaustivo, factorial y *backtracking*. Para los 2 primeros hay formulas exactas que nos dicen cuántos nodos se expandirán. Para *backtracking* no hay más respuesta que ejecutarlo. Sin embargo, usando una simulación de Montecarlo con un número apropiado de muestras se puede obtener un estimado de la cantidad de nodos visitados. Para esta tarea, $4 \leq N \leq 100$.

III. SALIDA

La salida de este proyecto será a consola en modo texto. Sin embargo, se espera que el despliegue esté formateado apropiadamente con calidad profesional.

Se mostrarán los resultados desde $N = 4$ hasta $N = 100$. En cada fila habrá 7 columnas que serán:

1. Cantidad de reinas.
2. Nodos visitados por el algoritmo exhaustivo.
3. Nodos visitados por el algoritmo factorial.
4. Nodos visitados por el algoritmo de *backtracking*.
5. Nodos *promising* hallados por el algoritmo de *backtracking*.
6. Nodos que sean solución hallados por el algoritmo de *backtracking*.
7. Estimado de nodos que visitaría el algoritmo de *backtracking* calculado por una simulación de Montecarlo.

Las columnas 2 y 3 se calculan con las fórmulas vistas en clase.

Para las columnas 4, 5 y 6 no hay más que correr el algoritmo de *backtracking*¹. Sin embargo, cada grupo debe determinar hasta que valor de N es razonable correrlo para que se pueda presentar en una demostración. Normalmente hasta $N = 15$ ó 16 toma un tiempo tolerable. Después de esa cantidad las columnas 4, 5 y 6 mostrarán “—”.

Trabajo extra opcional 1: tener lista una tabla corrida previamente donde se haya ejecutado *backtracking* hasta por lo menos $N = 20$, junto con las otras columnas.

La columna 7 mostrará el **promedio** de al menos 1000 ejecuciones de una simulación de Montecarlo del algoritmo de *backtracking*. Si desean hacer más ejecuciones, tome en cuenta que el tiempo de la demostración debe ser razonable.

Hasta donde sea posible, los números de todas las columnas se presentan en formato corriente, pero cuando sean cantidades demasiado grandes se usará notación científica. Recuerde que todo su despliegue debe estar formateado correctamente.

IV. REQUISITOS INDISPENSABLES

La ausencia de uno solo de los siguientes requisitos vuelve la tarea “no revisable” y recibe un 0 de calificación inmediata:

- La colaboración entre grupos se considera fraude académico.
- Utilizar código, archivos de datos, pruebas o documentación de semestres previos (e.g., buscando en GitHub) no sólo es fraude académico, es una aceptación de incompetencia e inutilidad. Es aún más patético si alguien hace trampa utilizando ChatGPT.
- Todo el código debe estar escrito en C (no C++).
- El proyecto debe compilar y ejecutar en Linux. Todo debe estar **integrado**, explicaciones del tipo “*todo está bien pero no pudimos pegarlo*”² provocan la cancelación automática de la revisión.
- La presentación debe ser de mucha calidad.
- No debe dar “Segmentation Fault” bajo ninguna circunstancia.
- Hacer la demostración en una máquina que levante Linux de manera real (puede ser dual), es decir no usar máquinas virtuales.

V. FECHA DE ENTREGA

Revisiones a las 11:30 am el **Jueves 24 de Octubre**. Esta tarea cuenta por 2 tareas para el rubro de Tareas y Quices.

¹Obviamente, tienen que programarlo.

²esto incluye los supuestos casos cuando alguien del grupo de trabajo no hizo su parte – el profesor no está interesado en sus problemas de organización.