



Proyecto final - El maestro de ajedrez - Simulación computacional

Carlos Andres Delgado S, Msc *

19 Agosto 2019

Reglas

1. El proyecto debe ser entregado el **14 de Septiembre de 2019 08:00am** hora de Colombia, por el campus virtual. Se permiten entregas atrasadas con una sanción acumulativa de 15 % por hora o fracción. Por ejemplo, si entrega a las 08:00:01am se aplicará el retraso por una hora, si entrega a las 09:00:01am se le aplicará una sanción del 30 % y así sucesivamente. Por esta razón, entregue lo suficiente antes del plazo para evitar sanciones.
2. Se permiten grupos de hasta 3 estudiantes.
3. El informe debe ir en formato PDF. Los formatos editables (docx, odt, etc) tienen problemas a la hora de ver el archivo en diferentes versiones. El no cumplimiento de esta regla traerá sanciones a su proyecto.
4. Debe incluir en el informe los nombres completos y códigos de los integrantes del grupo. El no cumplimiento de esta regla traerá sanciones a su proyecto.
5. El código debe ir comentado explicando brevemente cada función y apartes importantes del código. Se debe indicar para que se utiliza en el algoritmo. El no cumplimiento de esta regla traerá sanciones a su proyecto.
6. Debe entregar el código fuente organizado en carpetas dentro del primer nivel del archivo comprimido, no cree una jerarquía compleja difícil de revisar. El no cumplimiento de esta regla traerá sanciones a su proyecto.
7. No se permite copiar código de Internet ni de sus compañeros. Si se encuentra código copiado de alguna parte el taller será anulado por completo.
8. Entregue un sólo archivo comprimido. No entregue archivos comprimidos dentro de archivos comprimidos, ya que esto dificulta la revisión enormemente. El no cumplimiento de esta regla traerá sanciones a su proyecto.
9. El informe debe tener buena presentación, esto hace parte de la nota de los puntos del proyecto
10. Este proyecto debe ser sustentado, la nota de la sustentación es individual y está entre 0 y 1, la cual se multiplica por la nota obtenida en el proyecto.
11. Es obligatorio el uso del lenguaje de programación Python y la librería Simpy.

* carlos.andres.delgado@correounivalle.edu.co

1. El maestro del ajedrez

Es el año 2030, usted se graduó de ingeniería de sistemas en el año 2020, pero desde el año 2026 cuando el robot NS-2 demostró que la IA fuerte era una realidad, el panorama laboral ha sido cambiado, los humanos con limitaciones de velocidad de escritura de código y de cansancio, además de su gran costo económico han sido reemplazados por estos robots. Usted tenía una vida ingrata programando en PHP, pero ganaba bien, ha sido despedido para ser reemplazado por una de estas unidades, aquí una foto del desgraciado:

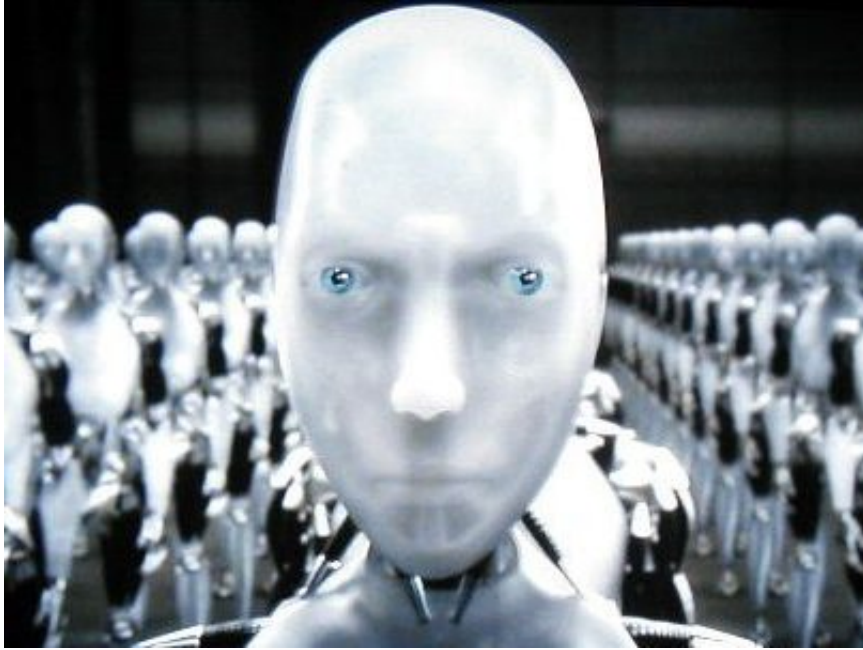


Figura 1: La foto del desgraciado

Lamentablemente, nunca escuchó a sus profesores y nunca aprendió a hacer otra cosa, como es el caso del estudiante que usted le hacía buying en la universidad que ahora es millonario, gracias que aprendió a ser analista de datos. Usted se encuentra muy preocupado porque debe responder por la alimentación de sus 4 hijos, 2 de los cuales se dedicaron a cantar reggaeton y usted sabe que los deberá mantener el resto de sus días.

Sin embargo, no está solo, el profesor de este curso también se quedó sin trabajo dado que un robot lo reemplazó y ahora la sede Tulua a excepción de los profesores nombrados que ahora trabajan como conserjes bien pagados, es manejada íntegramente por robots. Un día, usted se encuentra el profesor de este curso y le propone la idea de montar un negocio en el cual el profesor juega al juego de las n reinas con un robot. Si el humano gana obtiene 30, de los cuales 15 son para usted y si el robot gana, usted pierde 10.

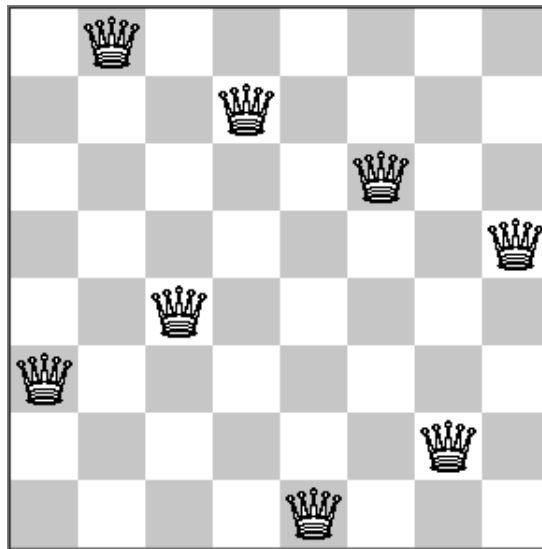


Figura 2: Ejemplo de ubicar 8 reinas ne un tablero 8 x 8

El problema de las n reinas consiste en colocar en un tablero de ajedrez $n \times n$, n reinas sin que estas se amenacen entre sí. Se tienen los siguientes datos:

1. Los robots llegan de acuerdo a una distribución uniforme entre 10 y 30 segundos.
2. El robot soluciona el problema de acuerdo con un algoritmo las vegas.
3. El profesor soluciona el problema de acuerdo con un algoritmo determinista.
4. Cuando va iniciar el juego, el robot y el humano seleccionan el tablero de acuerdo a una distribución uniforme. Los posibles valores de n son 4,5,6,8,10,12 y 15.
5. El primero que solucione el problema gana.
6. La simulación tiene una duración de 8 horas

Lo interesante del ejercicio es cómo incluir el tiempo de ejecución de un algoritmo como tiempo de servicio. Se busque que realice una simulación que ganancia obtiene usted después de este ejercicio.

2. Evaluación

Debe entregar un informe en formato PDF y la implementación usando la librería Simpy de Python.

1. **(25 puntos)** Documentación del modelo conceptual
 - Variables de estado, variables de desempeño, variables de entrada.
 - Especificar los eventos: Diagrama de flujo
 - Especificar proceso de simulación con un diagrama de flujo

2. **(30 puntos)** Implementación del modelo computacional

- Implemente en Simpy la simulación de este problema
- Explique en el informe cómo implemento este problema en la librería. Sustente que funciones utilizó y cómo manejó el flujo del programa.
- Diseñe una estrategia determinista para solucionar el problema de las n reinas.
- Diseñe una interfaz gráfica que permita ingresar el tamaño n del ajedrez. También agregue entradas necesarias para los escenarios que piensa diseñar.

Para esto entregue los archivos `simulacion.py`, `interfaz.py` y `main.py`. Se va ejecutar el archivo `main.py` el cual sólo debe contener código que llame los otros dos archivos.

3. **(15 puntos)** Compare la complejidad de la solución las Vegas con la solución determinista. Para esto evalúe valores de $n = 4, 5, 6, 8, 10, 12$ y 15 . Para esto puede usar una gráfica como la siguiente.

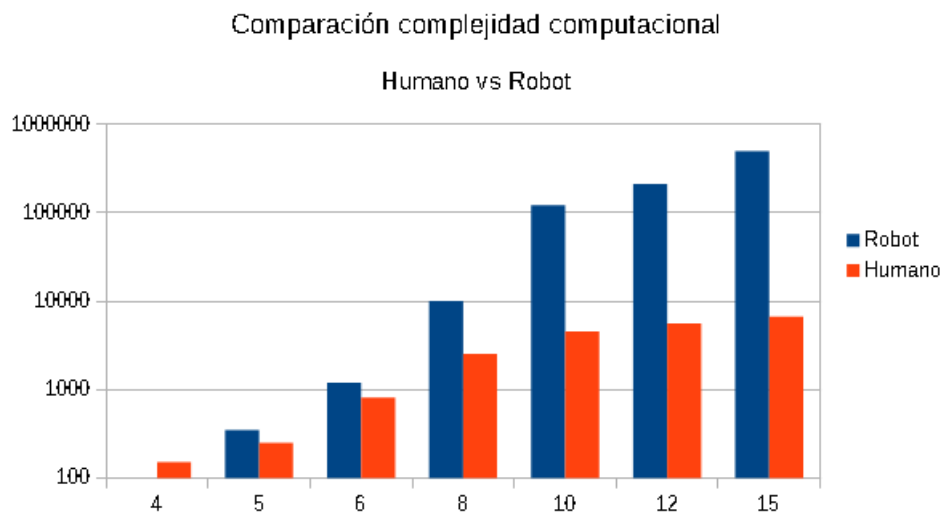


Figura 3: Comparación tiempo de ejecución humano y robot

¿A que cota $O(f(n))$ corresponde cada solución?. Para el algoritmos las Vegas corra 10 veces para cada valor de n .

4. **(20 puntos)** Diseñe 3 escenarios buscando mejorar la utilidad del negocio. Explique sus razones para el diseño de los escenarios e implementelos. Deje un archivo para cada escenario. `Escenario1.py`, `Escenario2.py` y `Escenario3.py`.
5. **(10 puntos)** De acuerdo a lo que encuentra ¿Que recomienda de acuerdo a su escenarios?, con su respectivo análisis.