

# Trabajo Práctico 1: Los Algoritmos Greedy son juegos de niños

El presente trabajo busca evaluar el desarrollo y análisis de un algoritmo Greedy. La fecha de entrega del mismo es el 19/09.

## Introducción y primeros años

Cuando Mateo nació, Sophia estaba muy contenta. Finalmente tendría un hermano con quien jugar. Sophi tenía 3 años cuando Mateo nació. Ya desde muy chicos, ella jugaba mucho con su hermano.

Pasaron los años, y fueron cambiando los juegos. Cuando Mateo cumplió 4 años, el padre de ambos le explicó un juego a Sophia: Se dispone una fila de  $n$  monedas, de diferentes valores. En cada turno, un jugador debe elegir alguna moneda. Pero no puede elegir cualquiera: sólo puede elegir o bien la primera de la fila, o bien la última. Al elegirla, la remueve de la fila, y le toca luego al otro jugador, quien debe elegir otra moneda siguiendo la misma regla. Siguen agarrando monedas hasta que no quede ninguna. Quien gane será quien tenga el mayor valor acumulado (por sumatoria).

El problema es que Mateo es aún pequeño para entender cómo funciona esto, por lo que Sophia debe elegir las monedas por él. Digamos, [Mateo está "jugando"](#). Aquí surge otro problema: Sophia es muy competitiva. Será buena hermana, pero no se va a dejar ganar (consideremos que tiene 7 nada más). Todo lo contrario. En la primaria aprendió algunas cosas sobre algoritmos greedy, y lo va a aplicar.

## Consigna

1. Hacer un análisis del problema, y proponer un algoritmo greedy que obtenga **la solución óptima** al problema planteado: Dados los  $n$  valores de todas las monedas, indicar qué monedas debe ir eligiendo Sophia para sí misma y para Mateo, de tal forma que se asegure de **ganar siempre**. Considerar que Sophia siempre comienza (para sí misma).
2. Demostrar que el algoritmo planteado obtiene siempre la solución óptima (desestimando el caso de una cantidad par de monedas de mismo valor, en cuyo caso siempre sería empate más allá de la estrategia de Sophia).
3. Escribir el algoritmo planteado. Describir y justificar la complejidad de dicho algoritmo. Analizar si (y cómo) afecta la variabilidad de los valores de las diferentes monedas a los tiempos del algoritmo planteado.
4. Analizar si (y cómo) afecta la variabilidad de los valores de las diferentes monedas a la optimalidad del algoritmo planteado.
5. Realizar ejemplos de ejecución para encontrar soluciones y corroborar lo encontrado. Adicionalmente, el curso proveerá con algunos casos particulares que deben cumplirse su optimalidad también.
6. Hacer mediciones de tiempos para corroborar la complejidad teórica indicada. Agregar los casos de prueba necesarios para dicha corroboración. Esta corroboración empírica debe realizarse confeccionando gráficos correspondientes, y utilizando la técnica de cuadrados mínimos. Para esto, [proveemos una explicación detallada](#), en conjunto de ejemplos.
7. Agregar cualquier conclusión que les parezca relevante.

## Entrega

Completar el [formulario de entrega](#) con los integrantes y el link al repositorio donde se encuentre el código fuente, y donde debe encontrarse el informe en formato PDF. Debe ser claro cómo ejecutar el programa pasando por parámetro un set de datos como los que se dan de ejemplo. No debe ser necesario tener la solución esperada (sino, ¿para qué programaríamos un programa para resolver un problema cuya respuesta ya conocemos?). Esto puede ser aclarado dentro del [README.md](#) del repositorio, u otra ubicación que les parezca clara.

El informe debe ser:

- Autocontenido: es decir, no debe ser necesario ponernos a buscar el código por diferentes lugares. El código que debe ser incluido es el código de los algoritmos a desarrollar (no es de interés un main, o el procesamiento de archivos), y estos deben estar incluidos donde consideren correcto dado el desarrollo del informe, para su entendimiento.
- Tener todo el análisis correspondiente.
- Ser realizado en un formato profesional. Para esto, les brindamos [un template](#) en  $\text{LATEX}$  para que puedan utilizar (también se encuentra en la home de la página del curso). No es necesario que lo sigan al pie de la letra, es simplemente un ejemplo que tiene varias cosas que pueden llegar a utilizar de  $\text{LATEX}$ . Si ya conocen  $\text{LATEX}$  no es necesario que lo utilicen, o mismo si utilizan algún otro formato (e.g. Markdown con Pandoc), pero recomendamos su revisión para que vean cosas que no deben de faltar. Por supuesto, pueden trabajar localmente como usar Overleaf o cualquier otra herramienta. El objetivo de darles el template no es la de limitar la creatividad, sino de asegurarnos que se cumplan lineamientos básicos sobre lo que se espera de una entrega de un informe en la facultad.
- En caso de ser necesarias reentregas, por favor agregar las modificaciones en un Anexo al final del informe. No modificar lo hecho anteriormente. La excepción a esto sería si hay que rehacer una enorme mayoría de lo escrito.

La nota del trabajo práctico tendrá en cuenta tanto la presentación y calidad de lo presentado, como también el desarrollo del trabajo. No será lo mismo un trabajo realizado con lo mínimo indispensable, que uno bien presentado, analizado, y probado con diferentes volúmenes, set de datos, o estrategias de generación de sets, en el caso que corresponda.