

Trabajo Práctico 2: Programación Dinámica For The Win

El presente trabajo busca evaluar el desarrollo y análisis de un algoritmo de Programación Dinámica. La fecha de entrega del mismo es el 14/10.

Introducción

Seguimos con la misma situación planteada en el trabajo práctico anterior, pero ahora pasaron varios años. Mateo ahora tiene 7 años. Los mismos años que tenía Sophia cuando comenzaron a jugar al juego de las monedas. Eso quiere decir que Mateo también ya aprendió sobre algoritmos greedy, y lo comenzó a aplicar. Esto hace que ahora quién gane dependa más de quién comience y un tanto de suerte.

Esto no le gusta nada a Sophia. Ella quiere estar segura de ganar siempre. Lo bueno es que ella comenzó a aprender sobre programación dinámica. Ahora va a aplicar esta nueva técnica para asegurarse ganar siempre que pueda.

Consigna

1. Hacer un análisis del problema, plantear la ecuación de recurrencia correspondiente y proponer un algoritmo por programación dinámica que obtenga la solución óptima al problema planteado: Dada la secuencia de monedas m_1, m_2, \dots, m_n , sabiendo que Sophia empieza el juego y que Mateo siempre elegirá la moneda más grande para sí entre la primera y la última moneda en sus respectivos turnos, definir qué monedas debe elegir Sophia para asegurarse obtener el **máximo valor acumulado posible**. Esto no necesariamente le asegurará a Sophia ganar, ya que puede ser que esto no sea obtenible, dado por cómo juega Mateo. Por ejemplo, para $[1, 10, 5]$, no importa lo que haga Sophia, Mateo ganará.
2. Demostrar que la ecuación de recurrencia planteada en el punto anterior en efecto nos lleva a obtener el **máximo valor acumulado posible**.
3. Escribir el algoritmo planteado. Describir y justificar la complejidad de dicho algoritmo. Analizar si (y cómo) afecta a los tiempos del algoritmo planteado la variabilidad de los valores de las monedas.
4. 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.
5. Hacer mediciones de tiempos para corroborar la complejidad teórica indicada. Agregar los casos de prueba necesarios para dicha corroboración (generando sus propios sets de datos). 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.
6. Agregar cualquier conclusión que parezca relevante.

Entrega

Debe enviarse al corrector asignado, por mail o slack, el link al repositorio donde se encuentre el código fuente, y donde debe encontrarse el informe en formato PDF, que debe seguir los lineamientos establecidos en el TP1. Debe ser claro cómo ejecutar el programa pasando por parámetro un set de datos como los que se dan de ejemplo. Esto puede ser dentro del `README.md` del repositorio, u otra forma que les parezca clara.

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.