

TRABAJO PRÁCTICO 3

PROBLEMA 1

El problema de 3-Coloreo es un problema de decisión: dado un grafo $G=(V,E)$, ¿existe una forma de colorear todos los vértices usando a lo sumo 3 colores, tal que cada par de vértices vecinos tengan diferentes colores?. Pero también tiene su contraparte como problema de optimización. Dado un grafo $G=(V,E)$, queremos colorear cada vértice con uno de tres colores, incluso si no se pudiera colorear con colores diferentes cada par de vértices vecinos. Diremos que una arista $e=(u,v)$ será *satisfecha* si los colores asignados a u y v son diferentes. Considerar un 3-Coloreo que maximiza la cantidad de aristas *satisfechas*, y sea c^* esta cantidad. Desarrollar un algoritmo aleatorio de tiempo polinomial que determine un 3-Coloreo cuyo número *esperado* de aristas *satisfechas* sea al menos $\frac{2}{3} c^*$ aristas.

Se pide:

- Diseñar e implementar un algoritmo aleatorio de tiempo polinomial.
- Diseño
 - Pseudocódigo
 - Estructuras de datos utilizadas
- Seguimiento: Ejemplo de seguimiento con un set de datos reducido
- Complejidad: Análisis de la complejidad temporal a partir del pseudocódigo
- Demostración: demostrar que el algoritmo cumple con el valor *esperado* mínimo solicitado
- Sets de datos:
 - Diseñar sets de datos para distintos grafos.
 - Se pueden generar utilizando una planilla de cálculo o cualquier herramienta que se considere adecuada.
 - Cada set de datos debe ser incluido en la entrega, junto con el resultado obtenido en cada caso.
 - El software entregado debe ser capaz de leer los sets de datos generados, procesarlos y dar el resultado solicitado
- Informe de Resultados: redactar un informe de resultados comparando los resultados de ejecución. ¿Se cumple con el valor *esperado* inicialmente?

PROBLEMA 2

En el problema de la mochila tenemos n ítems, cada uno con un peso w_i y un valor v_i . También tenemos un límite máximo W para el peso total de la mochila. El problema consiste en encontrar un subconjunto S de ítems que maximice el valor total V sujeto a la capacidad W de la mochila. En clase vimos que podemos diseñar un algoritmo que seleccione de manera aproximada los elementos del conjunto S cumpliendo con la capacidad W de la mochila, y que se aproxime al valor máximo óptimo con un margen $V/(1+\varepsilon)$. Pero en la vida real uno siempre quiere poner algo más dentro de la mochila, haciendo un poco de fuerza para cerrarla. Para un valor fijo $\varepsilon > 0$, implementar un algoritmo de aproximación que encuentre un subconjunto de ítems S que logre un valor de al menos V pero que se permita exceder la capacidad de la mochila en $(1+\varepsilon)W$.

Se pide:

- Diseñar e implementar un algoritmo de tiempo polinomial.
- Diseño
 - Pseudocódigo
 - Estructuras de datos utilizadas
- Seguimiento: Ejemplo de seguimiento con un set de datos reducido
- Complejidad: Análisis de la complejidad temporal a partir del pseudocódigo
- Demostración: demostrar que el algoritmo cumple con el criterio de aproximación solicitado
- Sets de datos:
 - Diseñar sets de datos para distintos conjuntos de elementos, valores y capacidades de la mochila.
 - Se pueden generar utilizando una planilla de cálculo o cualquier herramienta que se considere adecuada.
 - Cada set de datos debe ser incluido en la entrega, junto con el resultado obtenido en cada caso.
 - El software entregado debe ser capaz de leer los sets de datos generados, procesarlos y dar el resultado solicitado
- Informe de Resultados: redactar un informe de resultados comparando los resultados de ejecución. ¿Se cumple con el criterio de aproximación solicitado?