Proyecto de curso

Moderando el extremismo de opiniones en una red social

Análisis y Diseño de Algoritmos II

Escuela de Ingeniería de Sistemas y Computación

Profesor Juan Francisco Díaz Frias Profesor Jesús Alexander Aranda
Monitor Mauricio Muñoz

Agosto de 2024

1. Introducción

El presente proyecto tiene por objeto verificar que los estudiantes han adquirido el siguiente resultado de aprendizaje: Aplica técnicas de algoritmos voraces y dinámicos, en la construcción de algoritmos, para solucionar problemas de naturaleza combinatoria. Para ello, los estudiantes deben demostrar que logran:

- Aplicar cada estrategia de diseño (fuerza bruta, voraz y programación dinámica) a un ejemplo práctico.
- Usar un enfoque de fuerza bruta para resolver un problema y determinar si la estrategia conlleva a una solución óptima.
- Usar un enfoque voraz para resolver un problema y determinar si la estrategia conlleva a una solución óptima.
- Usar programación dinámica para resolver un problema comprendiendo que dicha estrategia conlleva a una solución óptima.
- Reconocer las ventajas y desventajas de utilizar diferentes estrategias de diseño (fuerza bruta, voraz y programación dinámica).
- Comparar distintas estrategias (fuerza bruta, voraz y programación dinámica) para un problema dado a partir del análisis del respectivo análisis de complejidad y optimalidad.

Para ello el estudiante:

- Desarrolla un programa utilizando un lenguaje de programación adecuado para resolver en grupo un proyecto de programación planteado por el profesor.

- Escribe un informe de proyecto, presentando los aspectos más relevantes del desarrollo realizado, para que un lector pueda evaluar el proyecto.
- Desarrolla una presentación digital, con los aspectos más relevantes del desarrollo realizado, para sustentar el trabajo ante los compañeros y el profesor.

2. El problema de moderar el extremismo de opiniones en una red social (ModEx)

2.1. El problema

Las redes sociales les ofrecen a sus usuarios fácil acceso de información y una amplia audiencia. Mientras estas redes facilitan el intercambio de ideas, a menudo conducen a la creación de cámaras de eco en las que los usuarios solo interactúan con quienes comparten sus opiniones, exacerbando así el extremismo. Este extremismo, que se observa en diversos temas, desde controversias triviales hasta mucho más profundas (por ejemplo, politica, economía etc), puede separar a las personas en bandos que tienen poca o ninguna comunicación y comprensión entre sí, y tiene un potencial efecto corrosivo y perjudicial en el funcionamiento de comunidades, sociedades y democracias. Por lo tanto, **es importante desarrollar mecanismos para reducirlo**. Esto se puede lograr invirtiendo recursos que permitan aumentar la conciencia y educar a las personas sobre los diferentes puntos de vista sobre un mismo tema, con el objetivo de moderar opiniones extremas y encontrar un terreno común más moderado. Este es un proceso arduo, complejo y costoso que puede tomar un tiempo significativo en dar resultados.

Este proyecto del curso es una simplificación del anterior problema.

Consideraremos una red social y su opinión con respecto a una determinada afirmación sobre un tema. Una red está formada por n agentes, cada uno los cuales tiene una opinión con respecto al tema, la cual se cuantifica en un valor entero entre -100 y 100, donde -100 indica una opinión totalmente desfavorable mientras 100 indica una opinión completamente favorable. Una opinión refleja un mayor nivel de moderación, entre más cercana esté a 0. Y entre más alejada esté la opinión de 0, esta refleja un mayor nivel de extremismo. Adicionalmente, cada agente tiene un nivel de receptividad o flexibilidad, el cual corresponde a un valor real entre 0 (mínima receptividad) y 1 (máxima receptividad), que modela cuán fácil o difícil es moderar la opinión del agente, siendo 1 la receptividad de un agente que modera su opinión con mucha facilidad y 0 la receptividad de un agente que modera su opinión con mucha dificultad.

El extremismo de la red depende del nivel de extremismo de cada uno de sus agentes. Lo que se quiere en este proyecto es reducir, lo más que se pueda, el extremismo de la red a partir de la moderación de la opinión de sus agentes.

Ahora, para poder moderar la opinión de un agente se necesita invertir un esfuerzo, cuyo valor depende de la opinión del agente y de su nivel de receptividad. Toda red cuenta con un valor máximo disponible, que se puede destinar para moderar la opinión de los agentes de la red, el cual se representa con un número entero positivo.

Con el presente proyecto se pretende abordar, a manera de ejercicio, el problema de determinar sobre cuáles agentes de la red se deben hacer los esfuerzos de moderación, para poder reducir el extremismo de la red en la mayor medida posible, respetando el valor máximo disponible para esta tarea. A este problema lo abreviaremos en adelante como *ModEx*

2.2. Formalización

Una red social $\mathcal{RS}$ es una pareja $< AG, R\_max >$, donde AG es una secuencia de agentes $AG =< a_0, \ldots, a_{n-1} >$, y $R\_max \geq 0 \in \mathbb{N}$ representa el valor entero máximo con que se cuenta para moderar las opiniones de la red RS.

Un agente a_i es una pareja $< o_i^{\mathcal{RS}}, r_i^{\mathcal{RS}} >$, donde $o_i^{\mathcal{RS}}$ representa la opinión del agente i de la red RS, y $r_i^{\mathcal{RS}}$ representa el nivel de receptividad del agente i de la red RS para $0 \leq i < n$.

El valor del extremismo de una red RS se puede definir de la siguiente manera:

$$Ext(\mathcal{RS}) = \frac{\sqrt[2]{\sum_{i=0}^{n-1}(o_i^{\mathcal{RS}})^2}}{n}$$

Una estrategia de moderación de una red $\mathcal{RS}$ de n agentes, es una secuencia $E =< e_0, e_1, \ldots, e_{n-1} >$ donde $e_i \in \{0, 1\}$; $e_i = 0$ indica que la opinión del agente a_i no se modera por medio de E, mientras $e_i = 1$ indica que la opinión del agente a_i se modera por medio de E.

Aplicar una estrategia de moderación E a una red $\mathcal{RS}$ de n agentes, denotado $Mod(\mathcal{RS}, E)$ da como resultado una nueva red $\mathcal{RS}'$ con la misma estructura de la red $\mathcal{RS}$, pero donde la opinión de los agentes moderados ha sido cambiada a 0, y la de los otros agentes no ha cambiado. Es decir: si $Mod(\mathcal{RS}, E) = \mathcal{RS}'$ entonces

$$o_i^{\mathcal{RS}'} = \begin{cases} o_i^{\mathcal{RS}} & \text{si } e_i = 0 \\ 0 & \text{si } e_i = 1 \end{cases}$$

El valor del esfuerzo de moderar la red RS con la estrategia E se define de la siguiente forma:

$$Esfuerzo(\mathcal{RS}, E) = \sum_{i=0}^{n-1} \lceil |o_i^{\mathcal{RS}} - o_i^{\mathcal{RS}'}| * (1 - r_i^{\mathcal{RS}}) \rceil$$

Una estrategia de moderación E sobre la red RS es aplicable si y solo sí el esfuerzo de moderar la red según dicha estrategia es menor o igual R_max, el valor máximo con que se cuenta para moderar las opiniones de la red RS.

El problema de la moderación óptima sobre una red social, ModEx, se define entonces así:

Entrada: Una red social $\mathcal{RS} =< AG, R\_max >$,

Salida: Una estrategia de moderación E aplicable para la red social $\mathcal{RS}$ (es decir tal que $Esfuerzo(\mathcal{RS}, E) \leq R\_max$), tal que $Ext(Mod(\mathcal{RS}, E))$ sea mínimo.

2.3. ¿Entendimos el problema?

2.3.1. Ejemplo 1

- Entrada: $\mathcal{RS}_1 = (<< 100, 0.5 >, < 100, 0.1 >, < -10, 0.1 >>, 55)$
- Estrategia de moderación E_1 de RS_1: $E_1 =< 0, 0, 1 >$

  Nótese que $Esfuerzo(\mathcal{RS}_1, E_1) = 0+0+\lceil 10*0.9 \rceil = 9 \leq 55$ por lo que la estrategia E_1 es aplicable.

  El nivel de extremismo de $Mod(\mathcal{RS}_1, E_1) = \mathcal{RS}_1'$ es

  $Ext(\mathcal{RS}_1') = \frac{\sqrt[2]{100^2+100^2+0^2}}{3} = 47.14$
- Estrategia de moderación E_2 de RS_1 : $E_2 =< 1, 0, 0 >$

  Nótese que $Esfuerzo(\mathcal{RS}_1, E_2) = \lceil 100 * 0.5 \rceil + 0 + 0 = 50 \leq 55$ por lo que la estrategia E_2 es aplicable.

  El nivel de extremismo de $Mod(\mathcal{RS}_1, E_2) = \mathcal{RS}_1''$ es

  $Ext(\mathcal{RS}_1'') = \frac{\sqrt[2]{0^2+100^2+-10^2}}{3} = 33.49$
- En este caso es mejor solución E_2 que E_1. ¿Habrá mejores? ¿Habrá estrategias de moderación no aplicables?

2.3.2. Ejemplo 2

- Entrada: $\mathcal{RS}_2 = (<< -30, 0.9 >, < 40, 0.1 >, < 50, 0.5 >>, 35)$
- Estrategia de moderación E_1 de RS_2: $E_1 =< 0, 1, 0 >$

  Nótese que $Esfuerzo(\mathcal{RS}_2, E_1) = 0 + \lceil 40 * 0.9 \rceil + 0 = 36 \not\leq 35$ por lo que la estrategia E_1 no es aplicable, por lo tanto independientemente de su nivel de extremismo no puede ser solución.
- Estrategia de moderación E_2 de RS_2: $E_2 =< 1, 0, 1 >$,

  Nótese que $Esfuerzo(\mathcal{RS}_2, E_2) = \lceil 30 * 0.1 \rceil + 0 + \lceil 50 * 0.5 \rceil = 28 \leq 35$ por lo que la estrategia E_2 es aplicable.

  El nivel de extremismo de $Mod(\mathcal{RS}_2, E_2) = \mathcal{RS}'_2$ es

  $Ext(\mathcal{RS}'_2) = \frac{\sqrt[2]{0^2+40^2+0^2}}{3} = 13.33$
- En este caso de las dos estrategías la segunda es la única aplicable ¿Habrá mejores aplicables?

3. El proyecto

Su proyecto consiste en explorar tres alternativas para solucionar el problema, una de fuerza bruta, una voraz y una usando programación dinámica, programarlas, analizarlas y hacer un informe al respecto.

3.1. Usando fuerza bruta

Considere el algoritmo que genera todas las soluciones posibles y entre ellas escoge la mejor.

3.1.1. Complejidad

¿Cuál es el orden de complejidad del algoritmo? Argumente su respuesta.

3.1.2. Corrección

¿El algoritmo siempre da la respuesta correcta al problema? Argumente su respuesta.

3.2. Usando un algoritmo voraz

3.2.1. Describiendo el algoritmo

Describa un algoritmo voraz para abordar el problema (ustedes como grupo deciden su estrategia voraz; la idea es que sea la más sencilla posible que ojalá les permita encontrar el óptimo siempre o la mayoría de las veces). **Se dará el puntaje máximo por este criterio a los mejores algoritmos voraces en términos de relación calidad de la solución / tiempo usado para encontrarla**.

3.2.2. Entendimos el algoritmo

Calcule la salida de su algoritmo para las entradas presentadas en 2.3.1 y 2.3.2 y calcule el costo de la solución. ¿Es la solución óptima?

Describa al menos 4 entradas más, calcule la solución entregada por el algoritmo y verifique si es o no la óptima. Guarde los resultados en una tabla.

3.2.3. Complejidad

¿Cuál es el orden de complejidad del algoritmo? Argumente su respuesta.

3.2.4. Corrección

¿El algoritmo siempre da la respuesta correcta al problema? Argumente su respuesta. Si la respuesta es negativa, adicionalmente analice cuándo si y cuándo no da la respuesta correcta.

3.3. Usando programación dinámica

Una alternativa a las dos anteriores sería utilizar programación dinámica. Para hacerlo es necesario comprobar que el problema tiene la propiedad de subestructura óptima y dejarse guiar por ella.

3.3.1. Caracterizando la estructura de una solución óptima

Caracterice la estructura de una solución óptima y a partir de ella describa el conjunto de subproblemas para los cuáles será necesario calcular las soluciones óptimas. ¿Cuántos subproblemas hay?

3.3.2. Definiendo recursivamente el valor de una solución óptima

Defina recursivamente el costo de una solución óptima para cada subproblema definido en el punto anterior.

3.3.3. Describiendo el algoritmo para calcular el costo de una solución óptima

Describa el algoritmo que calcula el costo de una solución óptima al problema original, basado en el punto anterior.

3.3.4. Describiendo el algoritmo para calcular una solución óptima

Tome el algoritmo descrito en el punto anterior y adicione:

- Una estructura que permita almacenar datos para construir una solución óptima cuyo valor sea el calculado por el algoritmo.
- Un algoritmo que recorra esa estructura construyendo una solución de costo óptimo.

3.3.5. Complejidad

Complejidad en tiempo. Calcule la complejidad en tiempo del algoritmo en términos de n.

Complejidad en espacio. Calcule la complejidad en espacio del algoritmo en términos de n.

¿Es útil en la práctica? Suponga que tiene un equipo que procesa 3×10^8 operaciones por minuto y que utiliza 4 bytes de almacenamiento por celda.
Si $n = 2^k$:

- Argumente para qué valores de k el tiempo que se tomará el equipo en resolver el problema es aceptable o inaceptable en la práctica.
- Argumente para qué valores de k el espacio que se tomará el equipo en resolver el problema es aceptable o inaceptable en la práctica.

[**Sugerencia:**] Utilice las siguientes aproximaciones:
$n+1 \sim n,\ A+1 \sim A$
1 minuto $= \frac{1}{3^4 2^8 5^2}$ años
1 año $= 3^4 2^8 5^2$ minutos $= 518400$ minutos
2^{10} Bytes $= 1$ KB
2^{10} KB $= 1$ MB
2^{10} MB $= 1$ GB
10^3 MB ~ 1 GB

3.4. Implementación

Cada grupo debe entregar el código correspondiente a:

- En el ambiente de programación de su preferencia, un programa que contenga una función por cada una de las soluciones diseñadas y analizadas (fuerza bruta, voraz y programación dinámica). Cada función recibe una red social $\mathcal{RS}$ y devuelve una tripleta $(E, Esfuerzo(\mathcal{RS}, E), Ext(Mod(\mathcal{RS}, E))$ que contiene E, la respuesta al problema, y el valor del esfuerzo de esa solución y del extremismo de la red moderada. Esas funciones en su código deben llamarse `modexFB, modexV` y `modexPD`.
- Usando la tecnología de su preferencia, una interfaz que permita leer entradas al problema desde un archivo de texto en un formato especificado, y escribir las salidas al problema en un archivo de texto en un formato especificado, así como visualizar las entradas, después de leerlas, y las salidas después de ser calculadas por cualquiera de las funciones solicitadas. Las salidas deben poder ser revisadas por medio de la interfaz en cuanto a su estructura y calidad (optimalidad). En particular debe ser fácil comprobar la corrección del costo de la solución dada.

Todos los programas deben correr bajo un mismo ambiente; se debe entregar el programa **ejecutable** y el código fuente.

3.4.1. Formato de las entradas

Las entradas vendrán en un archivo de texto con $n+1$ líneas así:

```
n
op0, recep0
op1,recep1
.
.
.
op(n-1),recep(n-1)
R_max
```

Es decir, la primera línea trae el número de agentes de la red social (un entero n); las siguientes n líneas traen la opinión y el nivel de receptividad de cada uno de los n agentes, y la última línea trae el valor máximo disponible para cambiar opiniones en dicha red.

3.4.2. Formato de las salidas

Las salidas se deben producir en un archivo de texto con $n+1$ líneas así:

```
Ext
Esf
mod0
mod1
mod2
.
.
.
mod(n-1)
```

Es decir, la primera línea trae el extremismo de la red una vez moderada usando la estrategia escogida, la segunda línea tiene el esfuerzo para llevar a cabo dicha estrategia; luego vienen n líneas, indicando sí cada uno de los agentes efectivamente fue moderado o no, es decir indicando la estrategia que lleva a la solución óptima.

3.5. Entrega

La entrega se debe realizar vía el campus virtual en las fechas previstas para ello, por uno sólo de los integrantes del grupo.

La fecha de entrega límite es el 26 de septiembre de 2024 las 23:59. La sustentación será en las sesiones del 3 y 8 de octubre de 2024.

Debe entregar un informe en formato pdf, los archivos fuente, un archivo Readme.txt que describa todos los archivos entregados y las instrucciones para ejecutar la aplicación. Todo lo anterior en un solo archivo empaquetado cuyo nombre contiene los apellidos de los autores y cuya extensión corresponde al modo de compresión. Por ejemplo ArandaDiazMuñoz.zip o ArandaDiazMuñoz.rar, o ArandaDiaz-Muñoz.tgz o ...

4. Sustentación y calificación

El trabajo debe ser sustentado por los autores en el día y hora especificado para su grupo. **La calificación del proyecto para el grupo** se hará teniendo en cuenta los siguientes criterios:

1. Informe (1/2) Debe responder a cada una de las solicitudes formuladas en el enunciado y debe contener al menos una sección dedicada a cada tipo de algoritmo implementado (fuerza bruta, voraz, programación dinámica), además de una sección dedicada a comparar los resultados de las tres soluciones implementadas (para esto use tablas y diseñe casos de prueba y documente cómo los diseñó) y una sección dedicada a concluir sobre las ventajas y desventajas de usar en la práctica los diferentes enfoques.

2. Implementación (1/2). Esto quiere decir que el código entregado funciona por lo menos para las diferentes pruebas que tendremos para evaluarlo y corresponde con todo lo solicitado (incluyendo la interfaz de lectura y visualización solicitada, y la generación de casos de prueba)

En todos los casos la sustentación será pilar fundamental de la nota individual asignada a cada integrante del grupo. En la sustentación se demuestra la capacidad del grupo de navegar en el código y realizar cambios rápidamente en él, así como la capacidad de responder con solvencia a las preguntas que se le realicen.

Cada persona de cada grupo, después de la sustentación, tendrá asignado un número real (el factor de multiplicación) entre 0 y 1, correspondiente al grado de calidad de su sustentación. **Su nota definitiva será la calificación del proyecto para el grupo, multiplicada por ese valor.** Si su asignación es

1, su nota será la del proyecto. Pero si su asignación es 0.9, su nota será 0.9 por la nota del proyecto. La no asistencia a la sustentación tendrá como resultado una asignación de un factor de 0.

La idea es que lo que no sea debidamente sustentado no vale así funcione muy bien!!! Y que, del trabajo en grupo, es importante que todos aprendan, no sólo algunos.

Éxitos!!!