Título del paper: debe ser un título descriptivo de lo que trata

Felipe Alejandro Villalobos Padilla

e-mail: felipe.villalobos.p@usach.cl

<enlace a alguna plataforma de interés de su estudio (github / linkedin / etc)>

*<Traducción fiel del título al inglés>.*

**RESUMEN:** *breve descripción de lo que trata el documento, debe incluir cuál es el problema(al menos solo el nombre del problema), como se trató (Que algoritmo se utilizó) y las conclusiones**(si el algoritmo es eficiente y con el orden obtenido). Está enfocado para llamar la atención del lector.*

**PALABRAS CLAVE**: palabras más frecuentes que se repiten a lo largo de su documento.

**ABSTRACT:** *traducción fiel del “RESUMEN” al inglés.*

**KEYWORDS.** *Traducción de “PALABRAS CLAVE” al inglés.*

# INTRODUCCIÓN

Se debe exponer el tema *(el tema es muy distinto al problema, hay que hacer la diferencia en esta parte ya que el problema surge del tema. Puede incluir imágenes)* que se tratará en el documento, porque se está tratando y que es lo que se espera encontrar como resultado (o que se espera aprender) y dar una pequeña información sobre el problema que se trabajará. ¿Entradas del problema?

# DESCRIPCIÓN DEL PROBLEMA

Describir con sus propias palabras el problema que se está tratando. Si es necesario, agregar una imagen explicativa, esta (y las demás) deben ir numeradas y con un título en la parte inferior.



*Figura 1: ejemplo de imagen explicativa.*

Note que si esta parte no queda clara, el resto del documento no tendrá ningún sentido ya que no se sabrá bien el problema que se está trabajando.

# DESCRIPCIÓN DEL MÉTODO

Lo más importante de esta parte es describir el tipo de algoritmo con el que se está trabajando (qué es y cómo funciona).

El algoritmo utilizado para la resolución del problema descrito en el capitulo anterior se conoce como volver atrás o Backtracking, es una estrategia de resolucion de problemas con alguna restriccion, donde el orden de la solucion no importa. Se dice que este es un algoritmo de busqueda en profundidad, debido a que en su funcionamiento durante la busqueda de la solucion, si se encuentra con una alternativa incorrecta, la busqueda retrocede hasta el paso anterior y toma la siguiente alternativa

# DESCRIPCIÓN DE LA SOLUCIÓN

Describir cómo se ha afrontado el problema, las suposiciones que ha hecho, herramientas utilizadas, etc. ¿Cuales fueron sus bases? ¿Que tomó como precondiciones? ¿Qué fue lo que obvió?

## REPRESENTACIÓN Y ESTRUCTURA

Se debe explicar cómo va a manejar la información del problema. Describir cómo va a tomar los datos del problema para interpretarlos como datos en la programación.

Debe hacer un listado con las estructuras que ha utilizado y una breve descripción de qué función cumplen en el código.

Debido al tipo de problema y a la solucion planteada, el manejo de datos sera llevado por matrices dinamicas, en especifico dos de estas, la primera de estas sera utilizada para realizar todas las operaciones de movientos en ella, tales como el posicionamiento de sucursales y el moverlas de lugar. Mientras se realizan estos movimientos la segunda matriz ira almacenando la representacion de la ciudad con mayor cantidad de sucursales.

## ORDEN DE COMPLEJIDAD

Siendo el mejor de los casos de orden O(n) debido a que este podria revisar solo una rama del arbol y llegar a la solucion ideal sin bifruxarse.

Pero en el peor de los casos se revisara todas las combinaciones posibles, es decir, todo el arbol de solucion haciendo que el orden de compjeidad ascienda a O().

Debido a que la forma de revisar Debe explicar qué complejidad tiene su

solución y cómo logró llegar a ese resultado.

(debe ser coherente con su pseudocódigo)

En esta parte es útil poner ecuaciones y utilizar las fórmulas vistas en clases. Por ejemplo si se tiene que el algoritmo propuesto tiene un tiempo de :

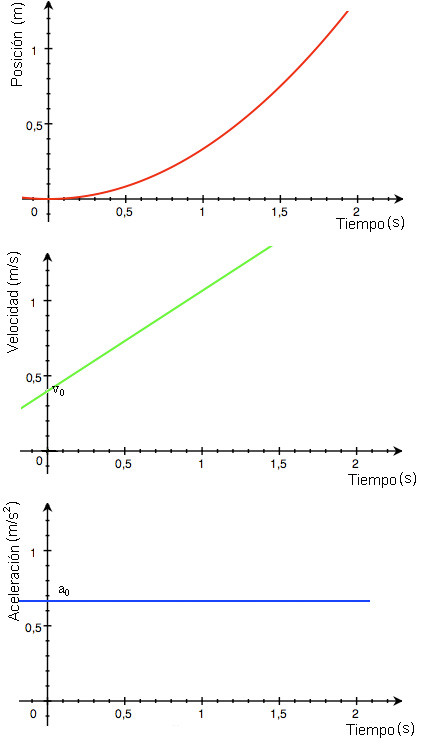
Entonces si luego se utiliza la fórmula para algoritmos recursivos por sustracción. El algoritmo finalmente tiene un orden de :

~

# ANÁLISIS DE LA SOLUCIÓN

En esta parte se debe hacer el estudio de cómo se ha implementado su solución.

Para todos los laboratorios es necesario incluir un gráfico de entradas vs tiempo y obtener una ecuación de ese gráfico (no se necesita un cálculo muy complejo, basta decir a qué función tiene parecida la curva)



*Figura 2: ejemplos de gráficos.*

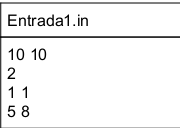
¿La ecuación se parece al orden obtenido anteriormente ? ¿ El número de entradas afecta en el tiempo de ejecución? ¿Qué ocurre si tengo 1000 entradas? ¿ Me sirve el algoritmo?

## ANÁLISIS DE IMPLEMENTACIÓN

Se deben responder las preguntas típicas para el análisis de un algoritmo (vistas en clases) . Las respuestas deben tener un fundamento, no sirve de nada poner un **SÍ** o un **NO** como respuesta. No es necesario poner respuestas muy extensas pero que sí expliquen cada punto ¿Porque es un algoritmo ?¿Por que para? ¿Para por que no hay bucles? ¿Es eficiente? ¿por que eficiente?. etc.

## EJECUCIÓN

Para la ejecución de este programa es necesario un archivo con la siguiente estructura:



donde la primera línea contiene dos números que representan el ancho y alto de la matriz respectivamente. En la siguiente línea irá un número entero N el cual representa las líneas siguientes que pueden haber. En las N líneas siguientes habrá un par de números por línea los cuales representan las posiciones de las sucursales (partiendo del 0) que deben

estar si o si.

Dado lo necesario para su funcionamiento, ahora se hablara de la compilación y ejecución, tanto para su forma normal, como su modo debug.

* Para compilar el programa:

$ make

* para ejecutar el programa:

$ ./lab.out archivo\_entrada

o

> lab.exe archivo\_entrada

Para la utilización del modo DEBUG y poder apreciar cada movimiento realizado se debe ejecutar de la siguiente manera:

ejemplo:

* Para compilar el programa en modo DEBUG:

$ make debug

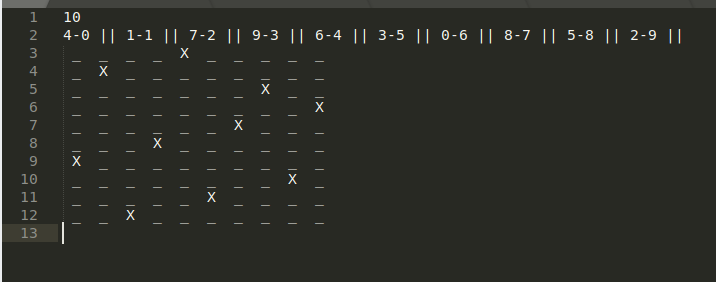
* para ejecutar el programa:

$ ./lab\_debug.out archivo\_entrada

o

> lab\_debug.exe archivo\_entrada

Dado la ejecución del programa, el archivo de salida para ambos casos cumple la siguiente estructura:



Donde la primera linea indica la cantidad de sucursales, la segunda linea muestra una lista con las coordenadas de las sucursales y las lineas siguientes una representación gráfica del mapa de la ciudad.

# CONCLUSIONES

Debe explicar las conclusiones que llego con este trabajo. que tan útil es el algoritmo desarrollado. ¿Fácil de implementar ? ¿Es recomendable utilizarlo? ¿Problemas de la implementación?

En los laboratorios 2 al 4, se debe hacer la comparación con los laboratorios anteriores. ¿Cual resulta mejor? ¿Porque resulta mejor?

Esta comparación puede ser sacada con la conclusión del paper del laboratorio anterior de su compañero.

# REFERENCIAS

Enlaces de donde se ha basado para el desarrollo del documento. El formato debe ser

* “<nombre del documento>” , <autor> ,<año> , disponible en : <enlace>

En caso de que alguno de estas partes no se encuentre, solo se debe poner “desconocido”

* “Plantilla Paper Lab Avanzados” , Nicolás Gutiérrez , 2018 , disponible en : [www.ejemplo.com/ngutierrez/Avanzados](http://www.ejemplo.com/ngutierrez/Avanzados)
* “ejemplo referencia”, Desconocido, 2010, disponible en : www.enlace.com