

LABORATORIO N°2

CURSO DE MÉTODOS DE

PROGRAMACIÓN

1-2020

CONTENIDO

Introducción	3
Estructuras de datos	3
Trabajo a realizar	6
Entregas y evaluación	8

1 INTRODUCCIÓN

En este documento se muestra el enunciado del segundo laboratorio a desarrollar dentro del curso de Métodos de Programación, el que busca evaluar los conocimientos adquiridos en la búsqueda de espacio de estados (BEE) y la búsqueda de espacio de soluciones (BES).

En este documento se indica lo que deberá desarrollar usted en este laboratorio. Las reglas que posee este trabajo y la forma de evaluación que tendrá. Finalmente se informan de las fechas y los elementos entregables, la forma de hacerlo llegar y otros elementos necesarios.

2 PROBLEMAS A RESOLVER

2.1 Problema usando búsqueda de espacio de estados

El problema de los misioneros y caníbales, también conocido como el problema de maridos celosos, son clásicos problemas de cruce de río (https://es.qwe.wiki/wiki/Missionaries_and_cannibals_problem).

El ejemplo base de este problema, es considerando que disponemos de tres misioneros y tres caníbales los que deben cruzar un río desde el lado A al B utilizando un bote, que solo puede transportar a dos personas a la vez y no puede moverse de un lado al otro si no tiene un pasajero como mínimo. La principal restricción es nunca dejar un número de caníbales mayor al de misioneros en el lado A, bote o lado B debido a que los caníbales se comerán al o los misioneros.

Una de las formas de solucionar el problema descrito anteriormente se detalla en la Figura 1, que indica los diferentes movimientos que deben realizar los misioneros y caníbales para llegar del lado A al B sin ninguno ser devorado.

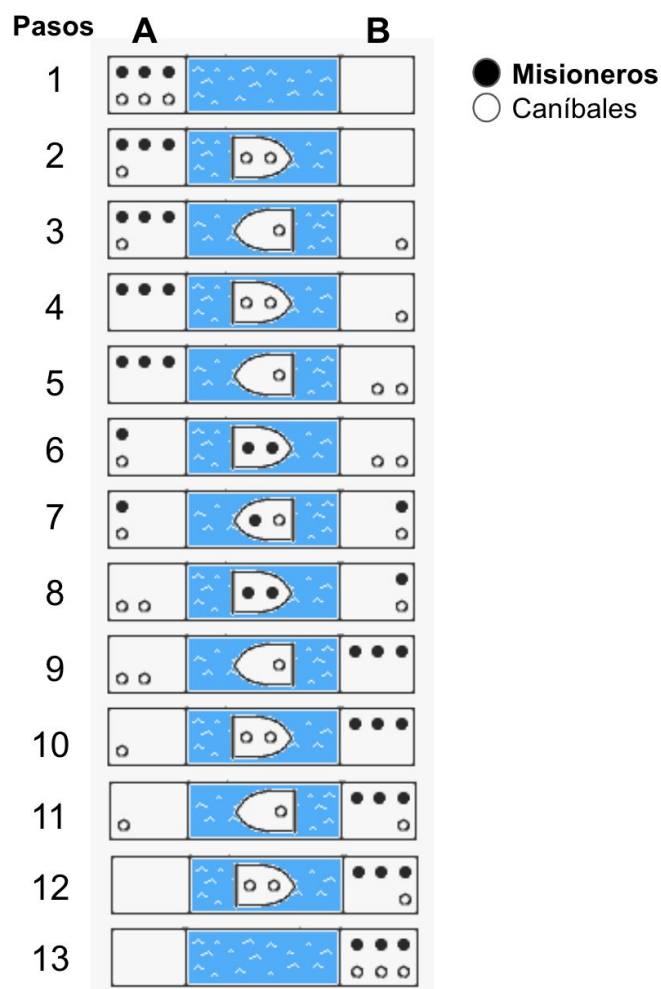


Figura 1: Resolución del caso base al problema de misioneros y caníbales (Imagen basada en <https://www.psicoactiva.com/puzzlecllopedia/misioneros-y-canibales/>).

Una forma para resolver el problema es analizando su comportamiento en un árbol y recorriendo sus nodos utilizando búsqueda en anchura, representando cada estado actual como un vector $\langle c \mid m \mid b \rangle$. El vector representa la cantidad de caníbales, misioneros y botes que se encuentran en el lado A o B. El ejemplo que se detalla en la Figura 2 muestra el estado inicial en el lado B y termina con el estado final en el lado B, pero este problema podría resolverse de la forma opuesta, es decir, partiendo con el estado inicial en el lado A y terminando con el estado final en el lado A.

Las acciones que se pueden realizar luego de cada estado son:

1. Mover un caníbal al lado opuesto donde está el bote.
2. Mover dos caníbales al lado opuesto donde está el bote.
3. Mover un misionero al lado opuesto donde está el bote.
4. Mover dos misioneros al lado opuesto donde está el bote.
5. Mover un misionero y un caníbal al lado opuesto donde está el bote.

Se considera que cualquier acción que genere como resultado un estado con una cantidad de caníbales mayor a la de misioneros en el lado A, B o bote no se considera como un estado posible y es por eso que no se detallan en la Figura 2. Además, si la ejecución de una acción nos lleva a un estado ya revisado en el lado B se detendrá y no se volverán a realizar posibles acciones en el estado.

Por ejemplo, partiendo del estado $\langle 0 \mid 0 \mid 0 \rangle$ los posibles estados que se obtienen utilizando las acciones respectivas son:

1. $\langle 1 \mid 0 \mid 1 \rangle$, es un estado posible debido a que la cantidad de misioneros es mayor a la de caníbales en el lado A.
2. $\langle 2 \mid 0 \mid 1 \rangle$, es un estado posible debido a que la cantidad de misioneros es mayor a la de caníbales en el lado A.
3. $\langle 0 \mid 1 \mid 1 \rangle$, **no** es un estado posible debido a que la cantidad de misioneros es menor a la de caníbales en el lado A.
4. $\langle 0 \mid 2 \mid 1 \rangle$, **no** es un estado posible debido a que la cantidad de misioneros es menor a la de caníbales en el lado A.
5. $\langle 1 \mid 1 \mid 1 \rangle$, es un estado posible debido a que la cantidad de misioneros es igual a la de caníbales en el lado A.

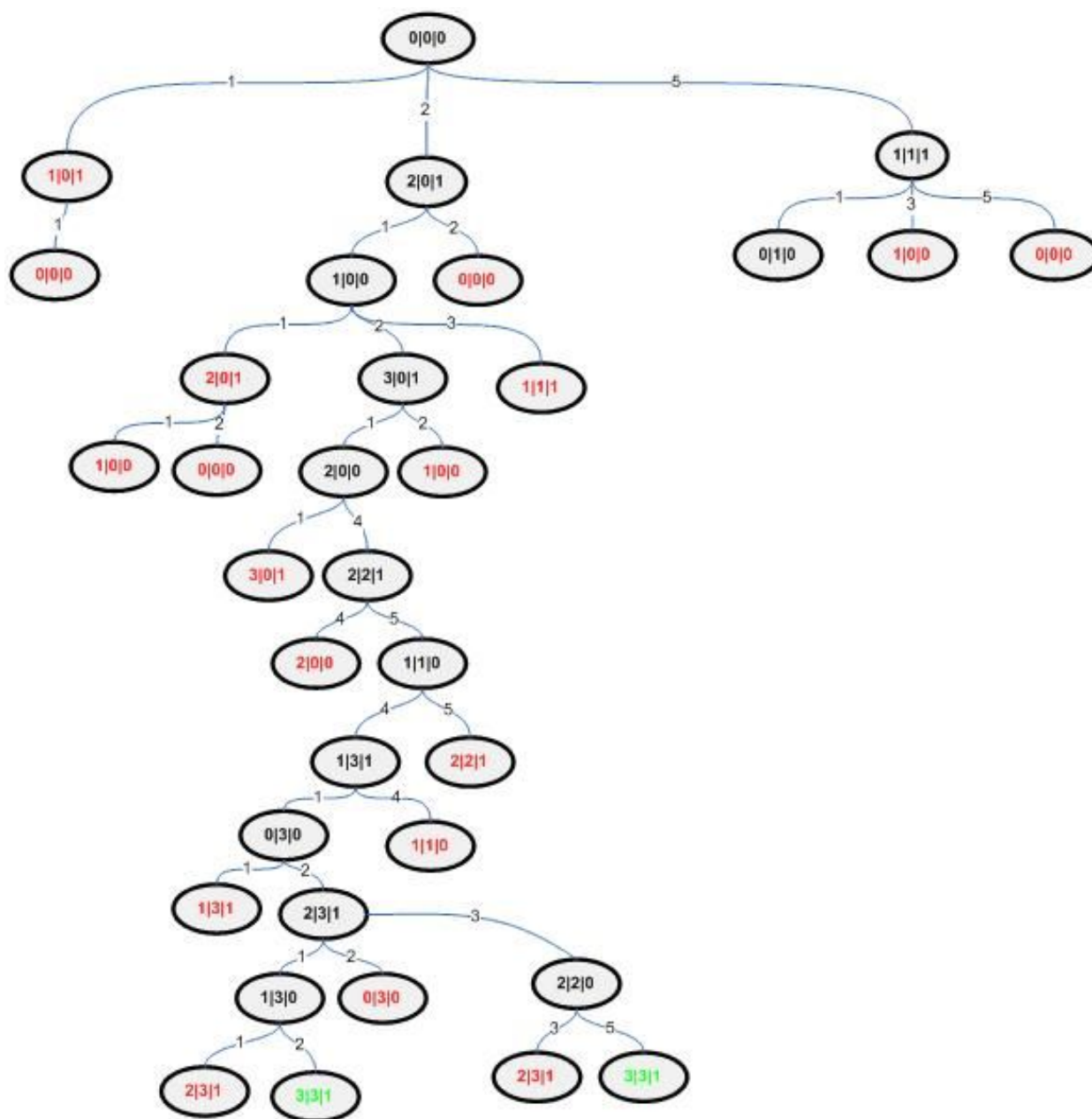


Figura 2: Resolución del caso base al problema de misioneros y caníbales, representado como árbol (Fuente: Agentes Inteligentes <http://hernandezland.blogspot.com/2008/12/problema-misioneros-y-canbales.html>).

2.2 Problema usando búsqueda en espacio de soluciones

El problema del cuadrado mágico consiste en, dada una matriz de $n \times n$, colocar números enteros positivos dentro de ella tal que la suma los números en las filas, columnas y diagonales sea el mismo. Por ejemplo, si $n=3$ entonces un cuadrado mágico vendría a ser un matriz de 3×3 con los números tal como se muestra en la siguiente Figura:

4	9	2
3	5	7
8	1	6

Figura 3: Ejemplo de cuadrado mágico 3×3

Cabe destacar que dado un tamaño n para el cuadrado mágico, existe más de una solución dependiendo de qué números se colocan dentro de este. La siguiente figura muestra otra solución para un cuadrado mágico de 3×3 :

22	8	21
16	17	18
13	26	12

Figura 4: Ejemplo 2 de cuadrado mágico de 3×3

En este laboratorio, usted deberá desarrollar un programa en C utilizando BES que permita generar los cuadrados mágicos de 3×3 con la restricción de que los números a colocar en cada casilla solamente pueden ser menores o igual a m , donde m es un valor ingresado por pantalla por el usuario. Tenga cuidado de utilizar para sus pruebas un valor de m pequeño (menor de 10) por el tiempo que pueda demorar. La generación de los cuadrados mágicos debe ser de manera automática. La salida de su programa debe ser a través de la terminal mostrando los resultados válidos de manera ordenada. Cabe destacar además que su programa deberá identificar los casos repetidos en que los números fueron, por ejemplo, traspuestos y solamente mostrar uno de ellos. La siguiente figura ilustra este caso:

22	8	21
16	17	18
13	26	12

22	16	13
8	17	26
21	18	12

Figura 5: Caso de cuadrados mágicos traspuestos

3 TRABAJO A REALIZAR

Debe implementarse en C las funciones que sean capaces de dar soluciones a los problemas descritos anteriormente. Si por alguna razón no logra disponer de un computador con el compilador gcc entonces existe la alternativa de usar la página (<https://www.onlinegdb.com/>) o (<https://coliru.stacked-crooked.com/>). Se advierte que si bien se puede editar, compilar, y ejecutar su código C usando dichas páginas, no hay garantía que funcione después usando gcc y por lo tanto solo usarlas en caso de real emergencia.

El programa principal debe mostrar inicialmente un menú con las siguientes opciones:

```
1) Canibales y misioneros
2) Cuadrado magico
3) Salir
Ingrese opcion:
```

En caso de ingresar una opción inválida debe imprimir “Opción inválida” y volver a imprimir el menú anterior. En el caso de escoger opción 1 debe solicitar al usuario ingresar la cantidad de misioneros y caníbales que se van a utilizar, el valor del caso base es 3. Obteniendo como resultado todos los posibles caminos que se tienen para solucionar el problema.

Ejemplo:

```
Ingrese numero de misioneros y canibales: 3
```

```
Solucion 1
```


<0|0|0>
<2|0|1>
<1|0|0>
<3|0|1>
<2|0|0>
<2|2|1>
<1|1|0>
<1|3|1>
<0|3|0>
<2|3|1>
<1|3|0>
<3|3|1>

Solucion 2

<0|0|0>
<2|0|1>
<1|0|0>
<3|0|1>
<2|0|0>
<2|2|1>
<1|1|0>
<1|3|1>
<0|3|0>
<2|3|1>
<2|2|0>
<3|3|1>

En caso de escoger 2 en el menú principal debe imprimir “Ingrese valor m: ”, el cual representa el máximo valor que puede ser colocado dentro de las celdas del cuadrado mágico. Y el resultado deben ser las soluciones de cuadrados mágicos posibles y el valor de la suma de estos, a continuación un ejemplo (m=26):

Solución 1

22 8 21
16 17 18
13 26 12

Suma: 51

Solución 2

4 9 2

3 5 7

8 1 6

Suma: 15

...

Y así hasta mostrar todas las posibles soluciones al problema. Recordar que el tamaño del cuadrado siempre será de 3x3 para este laboratorio.

Cuando el programa de las opciones 1 o 2 termine, entonces volver a presentar el menú inicial.

4 ENTREGAS Y EVALUACIÓN

La entrega consiste en la subida al portal del curso de Métodos de Programación, en el sitio www.udesantiagoovirtual.cl, en la actividad Entrega Laboratorio Individual N°2 el día 16 de Septiembre del año 2020 a las 23:55 horas.

En específico se deberá entregar un archivo comprimido en formato .zip o .rar que:

- Que tenga como nombre la siguiente estructura **<Sección Laboratorio>_<RUN>**:
 - o Primero debe ir la sección del laboratorio, por ejemplo, L23.
 - o Luego el RUN del creador del código, por ejemplo, 12345678-9
 - o Entonces el nombre final será L23_12345678-9
- El código fuente del programa en extensión .c
- Un archivo de texto plano llamado README, el cual debe indicar un paso a paso el cómo funciona su programa.

Para la evaluación, se revisan los siguientes puntos, con nota de 1 a 7 para cada uno, y el porcentaje correspondiente a cada punto está indicado entre paréntesis:

- Estructura obligatoria del programa (20%)
- Implementación general del programa (70%).
- Implementación correcta del menú (10%)
- Uso de buenas prácticas y estructura, en dónde:
 - o El código fuente debe tener como comentario, al inicio de él, la siguiente información:
 - RUN del creador del código.

- Sección del laboratorio a la cual pertenece el creador del código.
 - Fecha de creación del código.
 - o Las funciones creadas deben indicar:
 - Entradas: Qué significa cada una de las entradas de las funciones, por ejemplo, si se tiene una función suma que recibe dos números enteros, indicar que la entrada corresponde a los operandos de la operación adición.
 - Salida: Qué significa la salida generada.
 - Objetivo de la función.
 - o Uso de nombres de variables representativos.
 - o Uso de indentación y orden del código.
- Pruebas: Para corregir la implementación del algoritmo se somete el programa a pruebas.
 - o Dos prueba dadas por el equipo docente:
 - Las cuales consisten en las mostradas ya en este documento.
 - o Dos pruebas ocultas realizadas por el equipo docente:
 - Las cuales no serán mostradas hasta después del momento de revisar.
 - o Dos pruebas entregadas por el creador del código
 - Estas deben ir incluidas como ejemplos del archivo de entrada, dentro del archivo comprimido a subir.