

TAREA #10

Esta tarea tiene por objetivo que pongas en práctica la técnica de Backtracking y prepararte para el siguiente examen práctico.

La tarea consiste resolver los siguientes 2 problemas por medio de la técnica de Backtracking. Como parte del diseño de tu algoritmo de solución, responde las siguientes preguntas para cada problema y entrega la documentación de tus respuestas:

- ¿Qué representan los niveles del árbol de búsqueda de soluciones? ¿cuántos niveles tendrá el árbol?
- ¿Qué representan los hijos de un nodo en el árbol de búsqueda de soluciones? ¿cuántos hijos tendrá cada nodo?
- ¿En qué niveles se podrá encontrar la solución del problema?
- ¿Cuáles son las condiciones para que un nodo se expanda en el proceso de búsqueda de soluciones?

Acompaña la documentación con un dibujo del árbol, ejemplificando con un caso pequeño del problema.

PROBLEMA 1

En cierta escuela de la localidad, se está implementando un Centro de evaluación, el cuál consistirá de un salón en el que se aplicarán exámenes impresos o por computadora, según las necesidades del profesor que utilice el centro. El salón tiene capacidad para aplicar un examen simultáneamente a máximo 25 alumnos. La configuración del salón ha sido mapeada en una matriz de datos que guarda la distancia en centímetros que hay entre 2 mesas en las que los alumnos presentan el examen. Así, por ejemplo, el dato de la coordenada [5,8] sería la distancia en centímetros que hay entre la mesa 5 y la mesa 8 del salón. En el anexo, podrás encontrar la matriz a utilizar y que está identificada con el nombre de *distancias*.

Cuando un profesor aplica un examen en papel, la política del Centro es que el profesor entregue una versión diferente de examen a los alumnos que están en mesas que tienen un metro o menos de distancia.

Realiza un programa que sirva para indicarle al profesor usuario cuántas versiones mínimo diferentes de exámenes impresos debe de realizar para atender la política del Centro con la configuración de mesas que tiene en la matriz *distancias*.

Modifica también el algoritmo para que sólo genere una solución al problema, y no todas las combinaciones posibles.

PROBLEMA 2

Realiza la implementación de un programa que le pida al usuario un número entero N como entrada, y sirva para mostrar todas las posibles combinaciones de números enteros positivos que sumados dan como resultado el número N. Los números de cada combinación deberán ser todos diferentes y mostrarse en orden siempre ascendente.

Por ejemplo, si $N = 6$, las combinaciones a mostrar serían:

1+2+3
1+5
2+4
6

Observa que quedan fuera combinaciones como $3+3$ y $4+2$ por no cumplir las condiciones de números diferentes o estar en orden ascendente.

Para N = 10, las combinaciones a mostrar serían:

1+2+3+4
1+2+7
1+3+6
1+4+5
1+9
2+3+5
2+8
3+7
4+6
10

Entrega la impresión de los códigos fuente realizados, una impresión de las pruebas realizadas a cada problema, y un breve párrafo en el que narres cómo fue el proceso de solución de esta tarea, tus aprendizajes principales, y tus conclusiones respecto a la técnica de Backtracking.

El reporte de esta tarea se entregará el martes 23 de abril.

ANEXO: Matriz de datos para el problema 1-

```
int distancias[25][25] =
{{0, 88, 131, 74, 140, 115, 78, 127, 159, 144, 115, 139, 138, 108, 67, 133, 108, 193, 83, 185, 119, 114, 165, 107, 123},
{88, 0, 130, 175, 168, 140, 118, 157, 79, 160, 131, 89, 157, 121, 120, 97, 76, 108, 118, 153, 136, 119, 173, 83, 127},
{131, 130, 0, 114, 147, 67, 135, 145, 182, 133, 59, 107, 93, 115, 113, 129, 101, 100, 122, 175, 123, 123, 104, 82, 154},
{74, 175, 114, 0, 92, 194, 74, 153, 77, 115, 129, 117, 183, 155, 129, 80, 163, 95, 170, 103, 109, 148, 94, 52, 141},
{140, 168, 147, 92, 0, 99, 108, 173, 110, 108, 150, 130, 95, 118, 101, 174, 182, 102, 115, 141, 78, 115, 132, 96, 181},
{115, 140, 67, 194, 99, 0, 166, 135, 52, 130, 75, 64, 126, 147, 108, 128, 71, 128, 116, 189, 123, 145, 101, 163, 127},
{78, 118, 135, 74, 108, 166, 0, 108, 110, 123, 143, 115, 134, 121, 105, 69, 43, 51, 58, 57, 147, 108, 148, 110, 173},
{127, 157, 145, 153, 173, 135, 108, 0, 161, 126, 94, 62, 137, 144, 126, 144, 119, 139, 132, 149, 101, 175, 95, 145, 156},
{159, 79, 182, 77, 110, 52, 110, 161, 0, 96, 142, 177, 66, 80, 116, 94, 161, 112, 138, 54, 67, 114, 64, 95, 137},
{144, 160, 133, 115, 108, 130, 123, 126, 96, 0, 71, 103, 144, 81, 92, 114, 116, 145, 110, 110, 164, 166, 136, 133, 119},
{115, 131, 59, 129, 150, 75, 143, 94, 142, 71, 0, 110, 118, 128, 116, 101, 77, 94, 118, 134, 51, 60, 168, 70, 176},
{139, 89, 107, 117, 130, 64, 115, 62, 177, 103, 110, 0, 163, 109, 177, 69, 114, 93, 144, 97, 72, 113, 114, 106, 148},
{138, 157, 93, 183, 95, 126, 134, 137, 66, 144, 118, 163, 0, 144, 150, 168, 173, 115, 96, 109, 122, 116, 107, 156, 94},
{108, 121, 115, 155, 118, 147, 121, 144, 80, 81, 128, 109, 144, 0, 139, 146, 173, 123, 177, 91, 116, 101, 114, 153, 143},
{67, 120, 113, 129, 101, 108, 105, 126, 116, 92, 116, 177, 150, 139, 0, 159, 148, 88, 96, 131, 159, 149, 171, 97, 77},
{133, 97, 129, 80, 174, 128, 69, 144, 94, 114, 101, 69, 168, 146, 159, 0, 98, 96, 74, 145, 97, 100, 60, 84, 138},
{108, 76, 101, 163, 182, 71, 43, 119, 161, 116, 77, 114, 173, 173, 148, 98, 0, 72, 76, 130, 108, 171, 122, 117, 109},
{193, 108, 100, 95, 102, 128, 51, 139, 112, 145, 94, 93, 115, 123, 88, 96, 72, 0, 82, 91, 104, 81, 121, 115, 137},
{83, 118, 122, 170, 115, 116, 58, 132, 138, 110, 118, 144, 96, 177, 96, 74, 76, 82, 0, 129, 111, 133, 133, 110, 183},
{185, 153, 175, 103, 141, 189, 57, 149, 54, 110, 134, 97, 109, 91, 131, 145, 130, 91, 129, 0, 110, 85, 141, 163, 186},
{119, 136, 123, 109, 78, 123, 147, 101, 67, 164, 51, 72, 122, 116, 159, 97, 108, 104, 111, 110, 0, 129, 64, 66, 54},
{114, 119, 123, 148, 115, 145, 108, 175, 114, 166, 60, 113, 116, 101, 149, 100, 171, 81, 133, 85, 129, 0, 65, 155, 137},
{165, 173, 104, 94, 132, 101, 148, 95, 64, 136, 168, 114, 107, 114, 171, 60, 122, 121, 133, 141, 64, 65, 0, 129, 75},
{107, 83, 82, 52, 96, 163, 110, 145, 95, 133, 70, 106, 156, 153, 97, 84, 117, 115, 110, 163, 66, 155, 129, 0, 157},
{123, 127, 154, 141, 181, 127, 173, 156, 137, 119, 176, 148, 94, 143, 77, 138, 109, 137, 183, 186, 54, 137, 75, 157, 0}};
```