

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

## **CONTAINER PRE MARSHALLING PROBLEM**

**ERICK JHONATAN MAULEN TORO  
LUCAS TOMAS AGULLO VIVEROS  
MATIAS ALONSO ESPINOZA ARANGUIZ  
BENJAMIN JESUS ROJAS DELPINO**

ESTRUCTURAS DE DATOS AVANZADAS Y ALGORITMOS

Septiembre, 2020

# Índice

<b>Lista de Figuras</b>	<b>II</b>
<b>1 Modelado</b>	<b>1</b>
<b>2 Estrategia</b>	<b>2</b>
<b>3 Análisis</b>	<b>3</b>

## Lista de Figuras

1	Visualización de la primera heurística . . . . .	2
2	Visualización de la segunda heurística . . . . .	3

## 1. Modelado

Para el modelado del problema se decidió recrear un estado a través de una Matriz que contiene números de 0 a N, donde 0 sería un espacio vacío dentro del área restringida y cualquier número diferente de 0 , sería un container con M prioridad. Siguiendo las restricciones del problema, las transiciones de estados corresponden a las columnas de la matriz, las cuales se comportan como un Stack, es decir, solo se puede trabajar con el último dato ingresado o como se ve en el problema, el que está en la cima. El método del estado trabaja recibiendo dos parámetros, el primero el número de la columna de origen y el segundo el número de la columna de destino, por lo que el método quita el dato de más arriba de la pila (Top) y lo redirige al top de la columna de destino (Push). Las restricciones del método no permite hacer Push si la columna llegó al límite de altura del espacio, como también el número de origen y número de destino no puede ser el mismo y por último, la columna debe tener datos para realizar una transición.

## 2. Estrategia

Se utilizó un algoritmo de búsqueda *greedy*, el cual, por cada nivel, seguirá al nodo más prometedor, dejando de lado el resto. Los nodos serán evaluados en base a dos heurísticas.

1. **Numero de Ordenados:** La primera heurística consiste en contar la cantidad de containers que se encuentran ordenados, desde abajo hacia arriba. Esta será utilizada con el motivo de elevar el número de ordenados.

En la figura número 1 se encuentra un ejemplo, en dónde los containers encerrados en rojo son los que serán contabilizados para la primera evaluación. Los containers en la base de la matriz siempre serán contabilizados como ordenados, ya que se cuenta de abajo hacia arriba.

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

Figura 1: Visualización de la primera heurística

2. **Suma de Diferencias de Tops:** La segunda heurística corresponde a una suma de un coeficiente que representa la diferencia entre el container de la superficie de una pila y el container inferior a este, siempre y cuando este ordenado. Es decir, será una sumatoria de dicho coeficiente tomando en cuenta solo las pilas bien ordenadas. Mientras más grande es la diferencia, menor será el valor, esto es para darle prioridad a los movimientos en dónde se coloca el menor número posible sobre un stack.

En la ecuación número 1 podrán observar cómo se calcula la heurística y en la figura número dos, podrán ver, a partir de un ejemplo, cuales valores son los que se tomarán en cuenta.

$$Eval_2 = \sum \left( \frac{1}{(Stack_i[top] - Stack_i[top - 1]) + 1} \right) \quad (1)$$

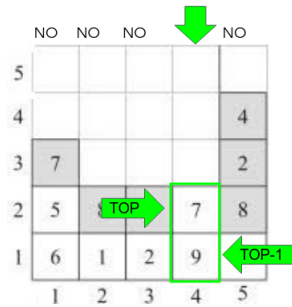


Figura 2: Visualización de la segunda heurística

### 3. Análisis

Se ocupó una matriz debido a su similitud espacial con el problema y además se contaba con la ventaja de tomar las columnas como un Stack, ya que solo se trabajara con el último dato ingresado. Debido a esto se ocupó la librería numpy, la razón radica en la cantidad de beneficios en el acceso de datos, principalmente en velocidad. Se implementó Greedy, debido a que se dieron resultados un poco más óptimos que el Best First y a largo plazo se penso que seria la mejor opcion.