

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

Desafío 1

Pablo Hurtado
Martín Toledo
Giselle Vásquez

SEPTIEMBRE, 2020

En este desafío se decidió por trabajar el problema bin-packing. En cual a partir de cierta cantidad de elementos con sus respectivos pesos deben ser almacenados a un contenedor de la manera mas óptima posible. Es decir, usando la mayor capacidad posible y así utilizar la menor cantidad de contenedores necesarios para guardar todos los elementos.

La suma de los pesos de los elementos del contenedor no deben superar el peso soportado por este. En un caso ideal, el contenedor debe estar ocupado en su máxima capacidad.

Para la resolución del problema, se hizo uso del lenguaje java y el IDE eclipse.

1. Resolución bin-packing

Para comenzar, se realiza la lectura de un archivo, el cual contiene en su primera linea, el peso máximo soportado por el contenedor(en este caso el contenedor es una maleta) y en la segunda linea los pesos de todos los elementos separados por espacios. Estos pesos pueden ser enteros o decimales.

Cómo la única información de interés relacionada a los objetos es su peso, no se consideró necesario crear una clase objeto para almacenarlos, sólo se creó un arrayList el cual guardó dicha información después de haberla leído del archivo.

Una vez que los pesos y el peso soportado por la maleta son almacenados, se crea un estado inicial, el cual consta de una maleta vacía (por convención del grupo se decidió comenzar rellenando de a una maleta y así optimizar los recursos utilizados).

Ya habiéndose creado el estado inicial, se comienzan a generar nuevos estados por nivel. Una vez generado un nivel se revisa y selecciona el mejor resultado. Cabe mencionar que para la generación de estados se realizó búsqueda en anchura y para su selección, best-first. Luego de seleccionar el mejor resultado se repite el proceso en un nuevo nivel hasta que todos los elementos que estaban almacenados en el arrayList fueron almacenados en maletas.

1.1. Generación, revisión y selección de la mejor opción

Para generar los estados de un nuevo nivel, se recorre el arrayList de objetos que fue rellenado en la lectura del archivo. Se evalúa si éste tiene un peso inferior o igual al espacio que tiene disponible la maleta. Si cabe en ella, se almacena y actualiza la capacidad de la maleta para seguir almacenando mas elementos en ella, además de generar un nuevo estado. En caso de que no se pueda guardar el objeto en la maleta, se sigue recorriendo el arrayList hasta encontrar un peso que quepa en ella o hasta haberlo recorrido completamente.

Si luego de haber recorrido todo el arrayList no se encontró ningún elemento que quepa en la maleta ó si la maleta esta llena, entonces se crea una nueva maleta en la cual se seguirán guardando nuevos elementos.

El proceso se repetirá hasta haber creado un nuevo estado para cada elemento que no ha sido guardado en una maleta.

Luego de haberse generado todos los estados posibles, se evalúan las opciones haciendo uso de una heurística conveniente, la cual compara entre todos los estados hasta encontrar el que haya dejado el menor espacio disponible, siendo este considerado como nodo padre para crear los posibles estados para el siguiente nivel.