UNCUYO
UNIVERSIDAD NACIONAL DE CUYO

FACULTAD DE INGENIERÍA

T.P. N° 1:

Búsqueda y optimización.

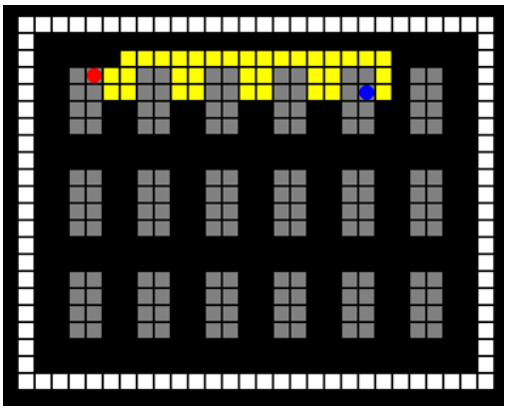
Alumnos:

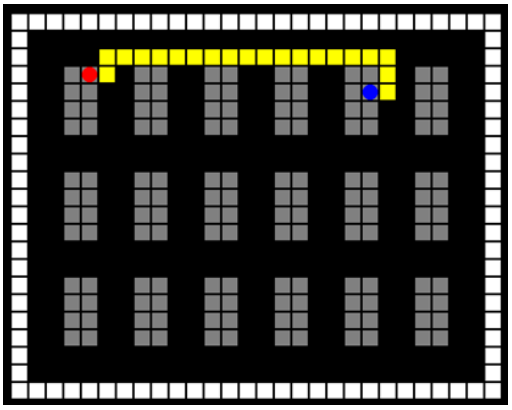
- Tomás Suárez
- Brenda Gudiño
- Esteban Vila

Ejercicio 1:

AlgorItmo A estrella.

El algoritmo reconoce el entorno (almacén) por sus posiciones que se encuentran agrupadas en frontera, pasillo y estantes.
Dadas 2 posiciones el algoritmo mira adónde puede avanzar, posiciones vecinas, calcula el costo de ir por cada una de ellas, selecciona la de menor costo y avanza repitiendo los pasos hasta llegar a destino.
Entorno: la construcción y visualización del almacén se consiguió con la librería turtle.
Para la resolución se crearon las funciones “hacer_almacen” a la cual se le pasa como argumento un string con la organización de la grilla y se analiza esa grilla para saber qué nodo pertenece a estantes, paredes, pasillos, inicio o fin.
Se definió la función “calcular_heuristica” que calcula la distancia en línea recta entre dos nodos, además se definió la función “calcular_costo” en la que se definió un costo de 1 para el movimiento en pasillos horizontales y 3 para el movimiento entre pasillos verticales porque si se tomaba para ambos el mismo costo el algoritmo tendía a irse siempre por los pasillos entre estanterías como se ve en la figura de la izquierda.

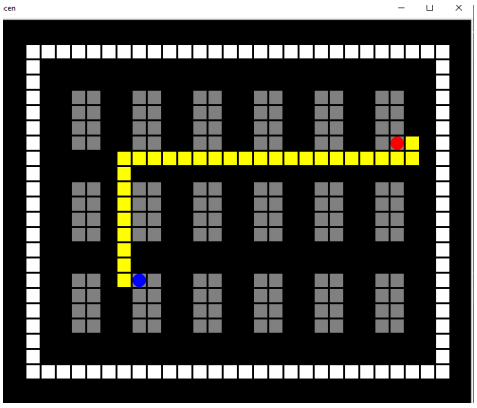


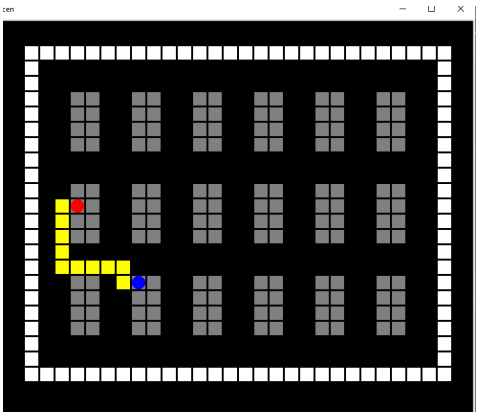


Otra función definida fue “buscar”, en ella se analiza quiénes son los nodos vecinos, se calcula su costo total, se actualiza la posición al nodo con menor costo total hasta llegar a la solución, por lo que al final se devuelve la solución y la distancia.

Por último se define una función para graficar la solución.

Le presentamos algunas pruebas del algoritmo.





En todos los ensayos en que se seleccionó correctamente a las estanterías, el algoritmo llegó a la solución.
Se concluye que este algoritmo es completo; siempre encuentra la solución y también es óptimo, además, tiene una complejidad de orden n.

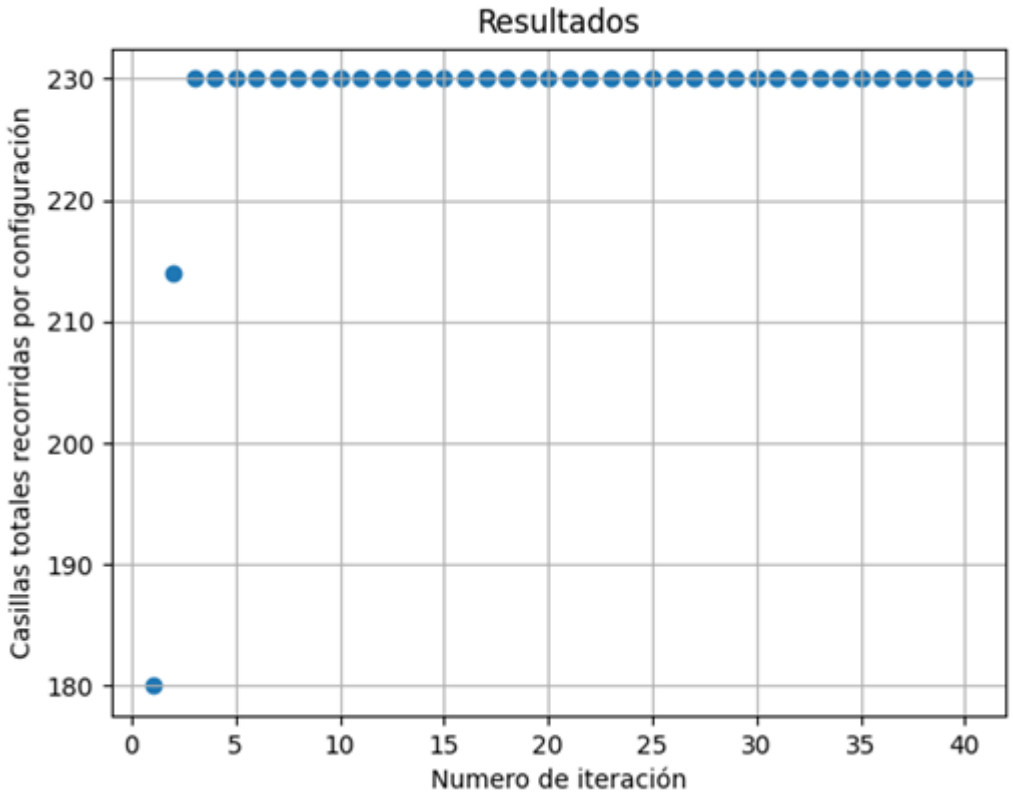
Ejercicio 2:

Algoritmo Temple simulado.

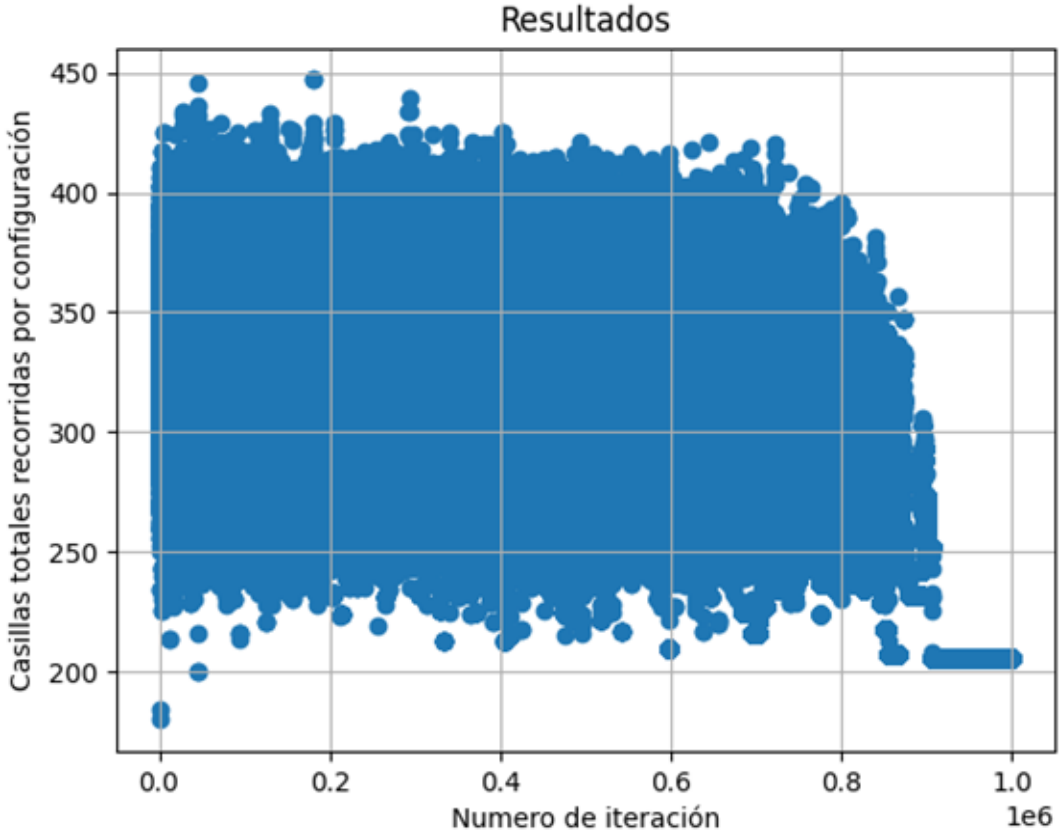
Para la implementación de este algoritmo primero se realizó una lectura de las órdenes y se las guardó en una lista. Luego se realizó la lectura de las distancias que se obtuvieron de la implementación del algoritmo A*.

Se creó la función “temple_simulado” que recibe como parámetros el arreglo de distancias, orden inicial, el valor de la T inicial y dos parámetros que definirán cómo será el enfriamiento de la temperatura. En esta función hay un bucle que se ejecutará mientras que la T actual sea mayor a 1e-10, aquí se actualizarán las variables según si el vecino que se escoja sea mejor o peor.

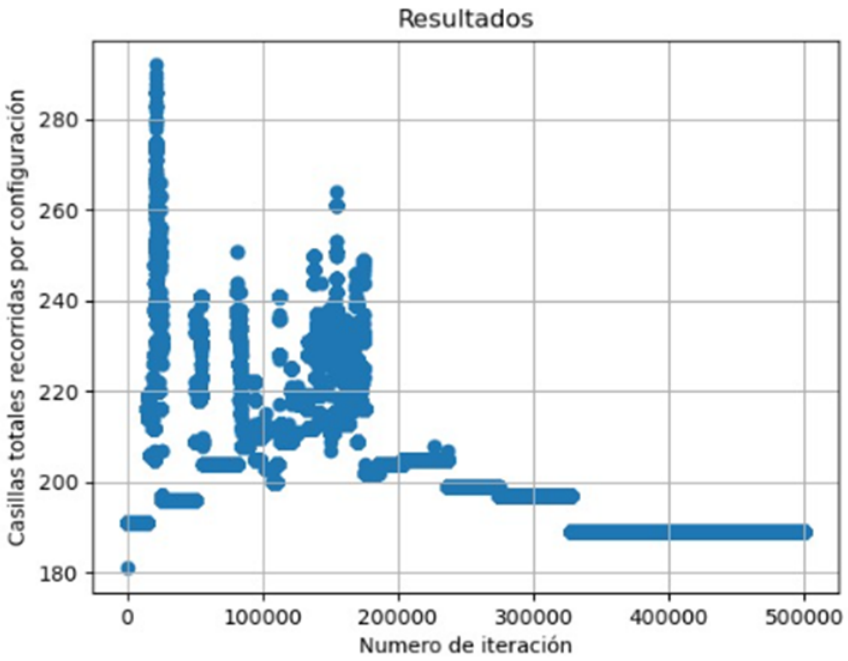
Por último se realizó la gráfica del historial en función del número de iteraciones y se analizó la tendencia del algoritmo según la modificación de los parámetros de temperatura y el tipo de enfriamiento, para lo que se pudo analizar lo siguiente.
Mientras mayor es la temperatura, más tiempo demora el programa ya que realiza mayor cantidad de iteraciones, lo contrario se produce si tomamos una temperatura menor.
Si se toma un enfriamiento exponencial a igual temperatura inicial realiza menor cantidad de iteraciones por ende dura menos tiempo
Por ejemplo a una T=100 para el enfriamiento logarítmico obtengo 40 iteraciones, 238 casillas recorridas y la siguiente gráfica.

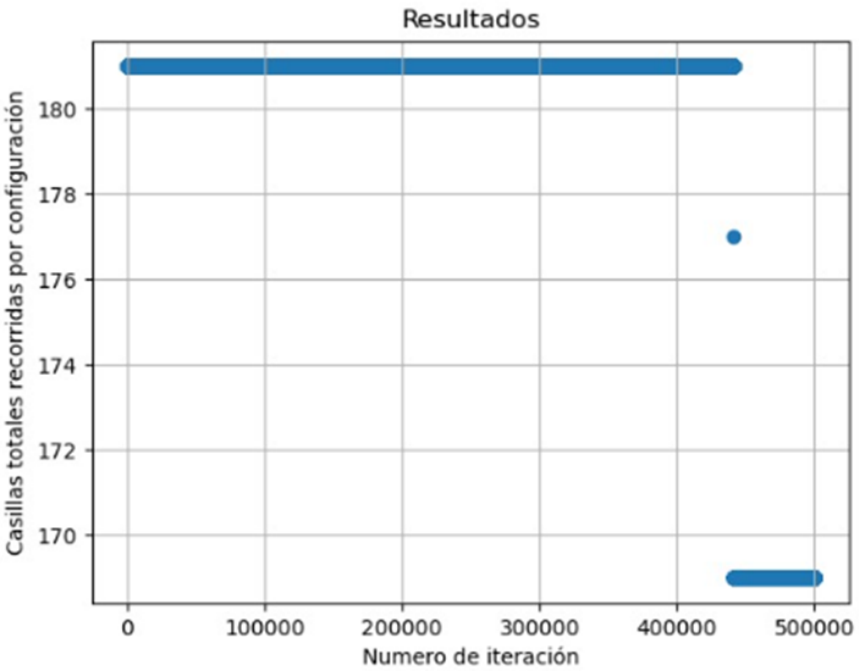


Por ejemplo a una T=100 para el enfriamiento lineal obtengo 100000 iteraciones y la siguiente gráfica



Para una T = 5 y con enfriamiento lineal, se obtienen las dos imágenes siguientes que se dan al ejecutar el programa dos veces.
La diferencia se debe a la aleatoriedad que involucra el desarrollo del problema.





Ejercicio 3.

Algoritmo genético.

A Continuación se presentan los resultado de una prueba del algoritmo:

```
Poblacion incial lista
==========================================================================================
------------------------------------ Algoritmo genético ------------------------------------
==========================================================================================
******************************************************************************************
---------------------------------------- Generacion 0 ----------------------------------------
******************************************************************************************
(Calculando fitness...)
Calculo fitness individuo 1 listo.
Fitness del individuo: 425.0
Calculo fitness individuo 2 listo.
Fitness del individuo: 409.0
Calculo fitness individuo 3 listo.
Fitness del individuo: 413.0
Calculo fitness individuo 4 listo.
Fitness del individuo: 425.0
Calculo fitness individuo 5 listo.
Fitness del individuo: 399.0
Calculo fitness individuo 6 listo.
Fitness del individuo: 385.0
Calculo fitness individuo 7 listo.
Fitness del individuo: 455.0
Calculo fitness individuo 8 listo.
Fitness del individuo: 357.0
Calculo fitness individuo 9 listo.
Fitness del individuo: 403.0
Calculo fitness individuo 10 listo.
Fitness del individuo: 385.0
(Seleccionando padres...)
(Realizando cruces...)
(Resultados)
Generacion: 0
```

```
Mejor individuo:

P75  P27    P138 P64    P100 P110   P77  P38    P129 P88    P63  P92
P44  P121   P74  P45    P116 P12    P106 P60    P124 P143   P130 P85
P112 P48    P42  P82    P76  P22    P104 P29    P117 P142   P91  P79
P3   P93    P28  P51    P115 P6     P125 P14    P69  P81    P94  P122


P128 P55    P95  P71    P35  P120   P126 P59    P53  P99    P2   P140
P16  P20    P80  P89    P111 P97    P31  P9     P56  P87    P83  P131
P58  P90    P67  P137   P141 P114   P13  P103   P25  P62    P32  P119
P24  P15    P134 P41    P19  P52    P66  P101   P10  P4     P33  P68


P49  P72    P18  P78    P132 P23    P98  P54    P118 P61    P108 P102
P1   P57    P40  P105   P21  P39    P30  P43    P133 P65    P139 P123
P135 P109   P86  P36    P50  P96    P70  P26    P34  P47    P5   P11
P127 P73    P113 P107   P46  P7     P84  P17    P144 P136   P37  P8



Fitness: 357.0
```

Pasos que sigue el algoritmo

- Se genera una población inicial aleatoria en la cual cada individuo de la población tiene los productos acomodados de una determinada forma según el archivo de texto provisto por la cátedra.
- Se calcula el fitness de cada individuo de la población por cada generación (o iteración).
- Luego se seleccionan los 2 mejores padres de la población de los cuales se obtienen dos hijos (individuos) nuevos, así se genera una lista que guarda esos hijos
- Luego se elige el mejor individuo según su fitness previamente calculado
- Se genera la nueva población con los hijos generados previamente y se realiza una mutación según una probabilidad definida.
- Esto se repite según la cantidad de generaciones que se hayan definido.
- Y por último se muestra el mejor individuo histórico y su fitness