

Procesador:	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.8GHz
Memoria:	16 GB

Tabla de tiempos medida con la clase 'TiemposColocacionSegmentos' para cada algoritmo:

n	tDEV1	tDEV1/nVeces	tDEV2	tDEV2/nVeces	tDEV3	tDEV3/nVeces
	milisegundos	segundos	milisegundos	segundos	milisegundos	segundos
100	97	0,097	228	0,228	112	0,112
200	154	0,154	377	0,377	227	0,227
400	325	0,325	781	0,781	444	0,444
800	679	0,679	1469	1,469	888	0,888
1600	1359	1,359	2836	2,836	1773	1,773
3200	2754	2,754	7609	7,609	3519	3,519
6400	5523	5,523	11076	11,076	7063	7,063
12800	11032	11,032	21670	21,67	14034	14,034

Teniendo en cuenta que se usó un nVeces = 1000000.

El algoritmo más eficiente según la medición de tiempos sería el devorador1.

Cuestiones:

El devorador2 sería el algoritmo que proporciona la solución óptima desde el punto de vista de la empresa para los ficheros 'juego1.txt' y 'juego2.txt', ya que proporciona el valor más alto de pufosos, Coste DEVORADOR2 = 507.5 pufosos y Coste DEVORADOR2 = 11599.5 pufosos respectivamente.

Sin embargo, el devorador3 sería el algoritmo que proporciona la solución óptima desde el punto de vista del jugador para los ficheros 'juego1.txt' y 'juego2.txt', ya que proporciona el valor más bajo de pufosos, Coste DEVORADOR3 = 170.5 pufosos y Coste DEVORADOR3 = 4660.5 pufosos.

Ninguno de los devoradores son el heurístico óptimo puesto que no dan el valor óptimo para cualquier problema dado.

Complejidades de los algoritmos devoradores:

Devorador 1:  $O(n)$ . Devorador2:  $O(n \log n)$ . Devorador3:  $O(n \log n)$ .

Los tiempos obtenidos se sincronizan con las complejidades dichas puesto que  $O(n) \ll O(n \log n)$ . El devorador2 posee la misma complejidad que el devorador3 pero tarda algo más de tiempo.