Comparación empírica entre algoritmo de fuerza bruta, backtraking O(n) y backtraking O(1)

Realizado por: Antonio Jesús Heredia Castillo

Los códigos implementados están junto a esta memoria, así como los datos y graficas obtenidas.

Implementación

Para mis algoritmos uso en el espacio de soluciones tuplas de tamaño variable. El árbol que utilizo se corresponde con aquel en el que los valores se encuentran en las aristas.

El algoritmo de fuerza bruta no es mas que un backtraking pero sin poda, es decir recorre todas las ramas del arbol buscando soluciones, en ningun momento descarta ninguna.

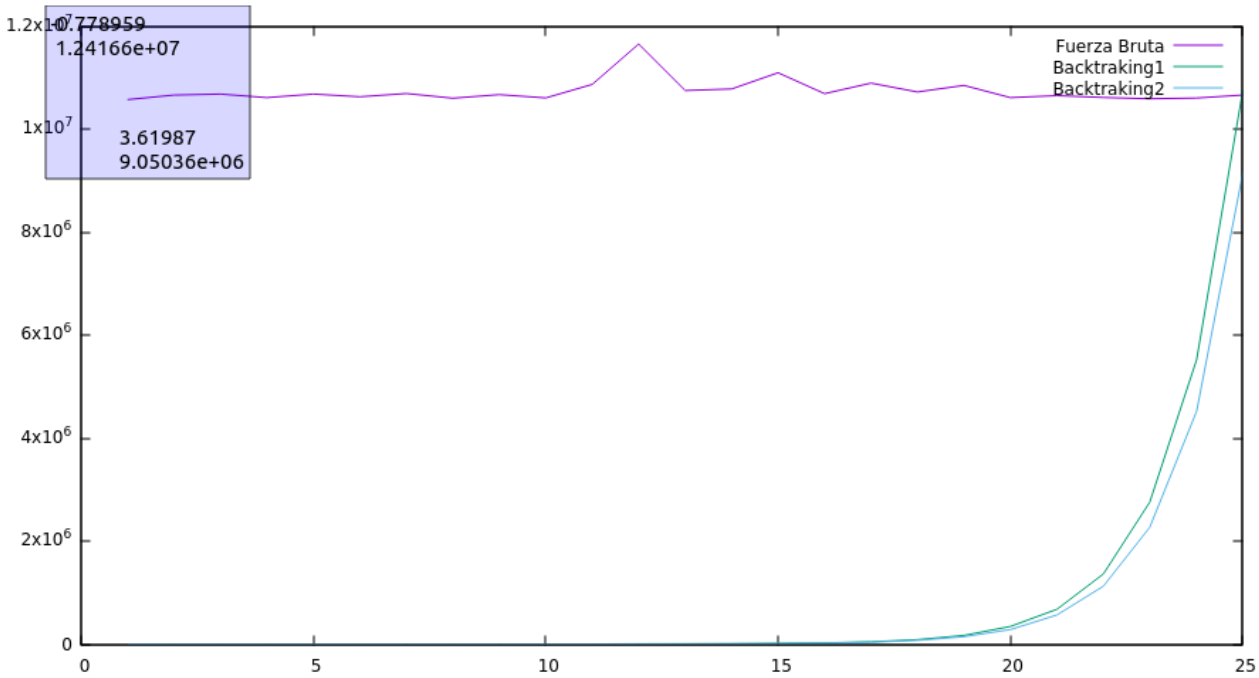
El primer algoritmo de backtraking usa una poda que consiste en que cuando el valor de la posición del vector Conjunto es mayor que la suma que queremos obtener, poda esa rama entera. Ya que no vamos a poder obtener ninguna solución ahi.

El segundo algoritmo de backtraking es una mejora del anterior, ya que para ver si una solución es factible no recorre todo el vector de solucion. Si no que lleva una suma guardada y la comprueba directamente, haciendo que sea O(1) y no O(n)

Por mi cuenta he añadido un tercer algoritmo de backtraking que es una mejora de los dos anteriores, ya que al tener la suma de la sulucion actual que esto probando, he añadido una mejora en la poda, haciendo que una vez que la suma sea mayor, pueda podar esa rama, ya que no se va a poder obtener una suma menor.

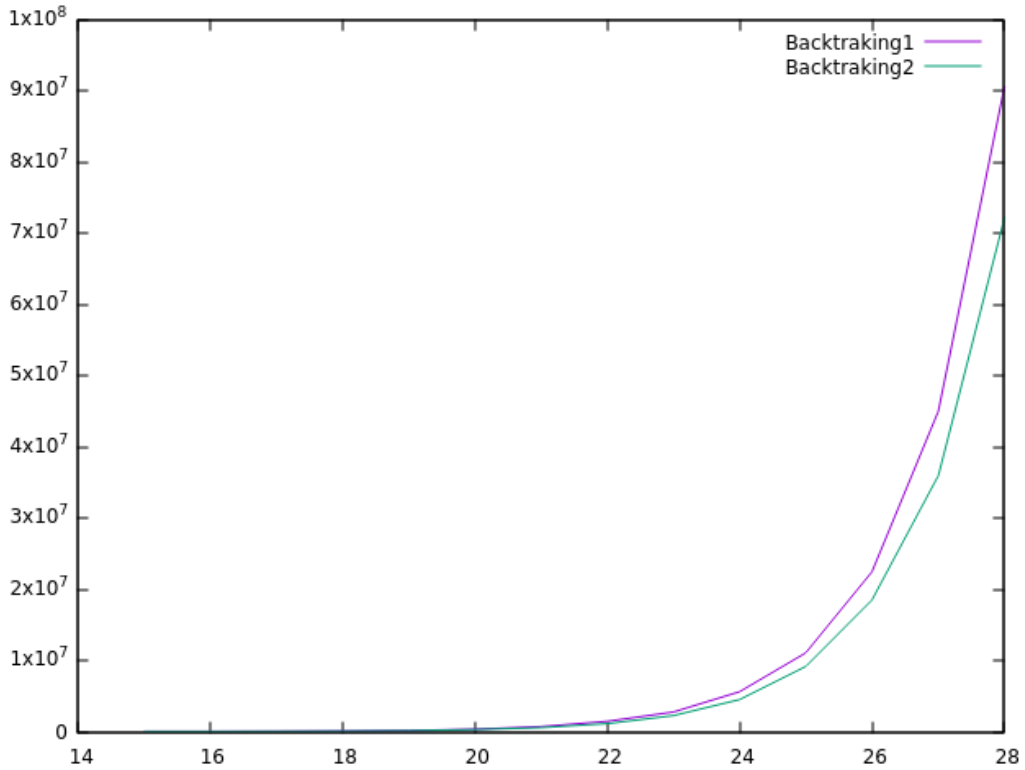
Estudio empirico

Esta grafica es la comparación entre los tres algoritmos que pedía la practica.



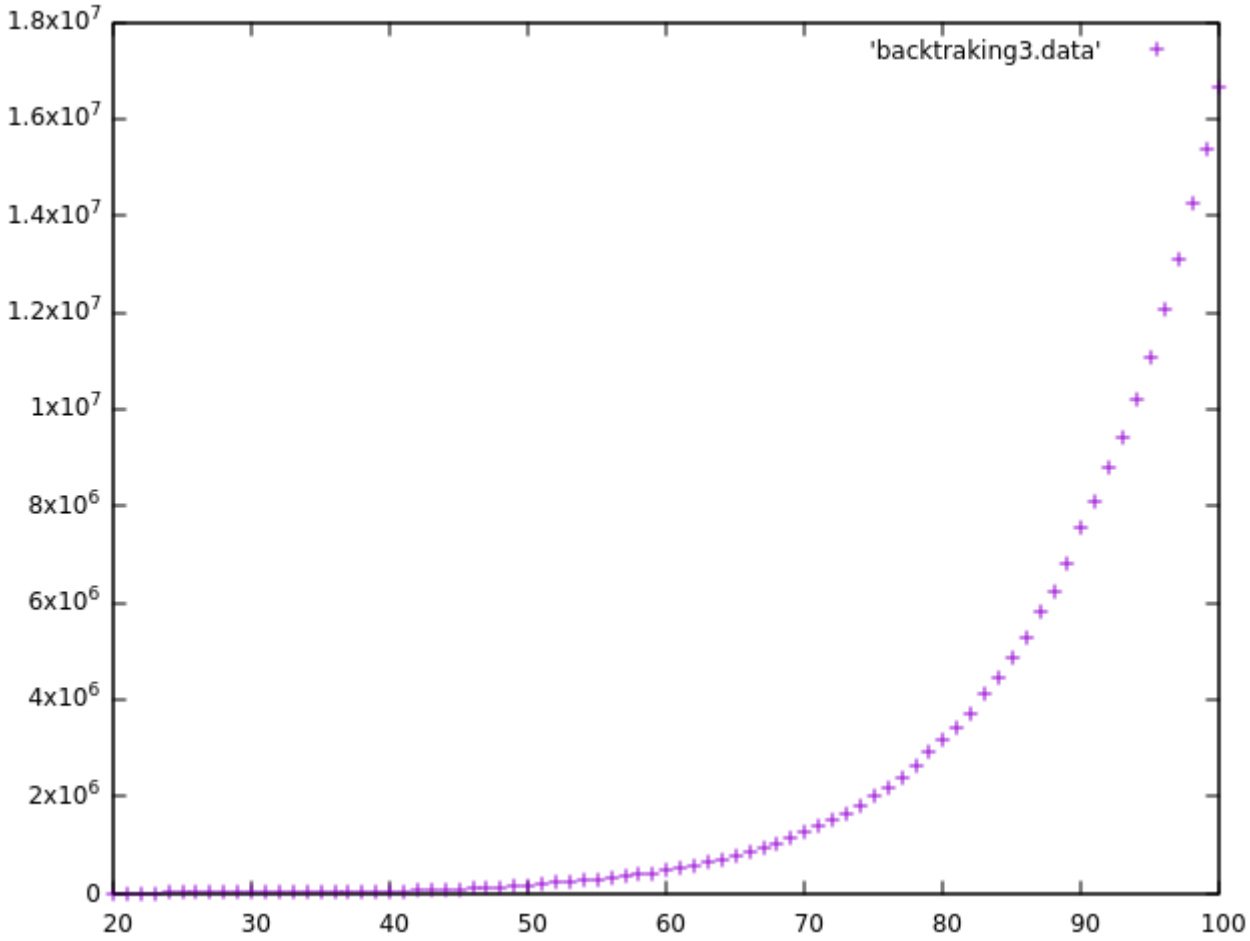
Como podemos observar el tiempo de fuerza bruta es siempre constante ya que en ningun momento realiza poda. En cambio los algoritmos de backtraking solamente se acerca al tiempo de fuerza bruta cuando la suma que queremos obtener es cercana al tamaño del conjunto, ya que no puede podar suficientes ramas para mejorar tiempo. Estos datos han sido obtenidos con un tamaño del conjunto de 25 elementos desde el 1 hasta el 26 y buscando una suma desde el 1 hasta el 25.

Podemos observar como el algoritmo O(1) es algo mas rapdio, aunque en esa grafica no se observa lo suficientemente bien. Para ello obtuve esta otra grafica.



Como podemos ver para suma 25 (que es el que se corresponde con la anterior grafica) parece no haber mucha diferencia pero con suma 27 la diferencia es de hasta 2 segundos menos. Aun asi tardando uno 9 segundos y otro 7 segundos

He obtenido mas datos para el algoritmo de backtraking 2 y el nuevo que he hecho con una poda mas efectiva, para ver como afectaba. Y mejoraba de forma muy sustancial. Ya que pasa de que en buscar el subconjunto de sumas hasta 35 no pudiera realizarla a realizar los subconjunto de sumas hasta 100 sin ningún problema en 1,6 segundos.



Así que si tuviera que elegir algun algoritmo ahora mismo me quedaría con la opción 3.