

ALGORITMIA

PRÁCTICA 6

Héctor Lavandeira Fernández
UO277303 | UNIVERSIDAD DE OVIEDO – CURSO 2021/22

Características principales del ordenador:

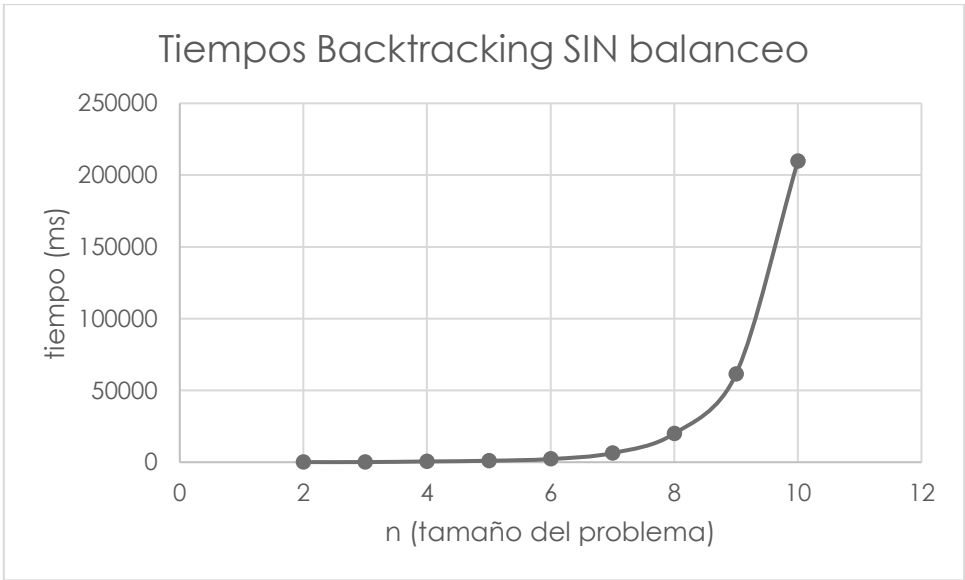
Procesador:	i5-8250U
Memoria RAM:	8GB

MEDIDAS DE TIEMPO

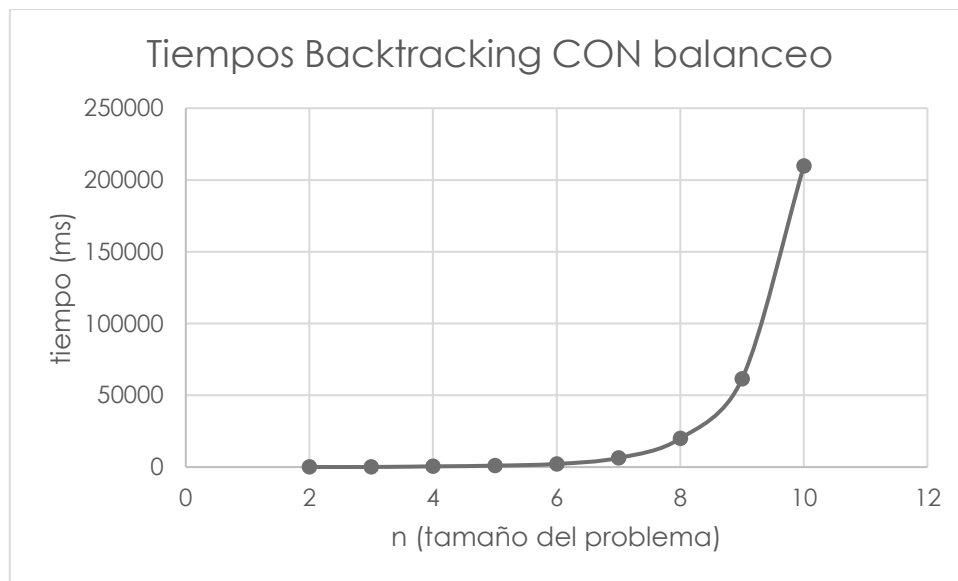
n	tiempo_bt (ms)	tiempo_bt_bal (ms)	zncc_voraz	zncc_bt	zncc_bt_bal
2	72	54	0.0	0.0	0.0
3	83	77	0.0	0.0305	0.0231
4	511	131	0.0	0.0245	0.0219
5	997	485	0.0243	0.0375	0.0375
6	2207	672	0.0219	0.0431	0.041
7	6357	1649	0.0406	0.0577	0.0563
8	19906	4236	0.06	0.0617	0.0566
9	61434	10341	0.0428	0.073	0.073
10	209753	27839	0.0323	0.0738	0.0725

NOTA: Para tamaños de problema superiores a 10, el tiempo de espera superaba los 15 minutos.

Representación de los tiempos de backtracking sin balanceo:



Representación de los tiempos de backtracking con balanceo:



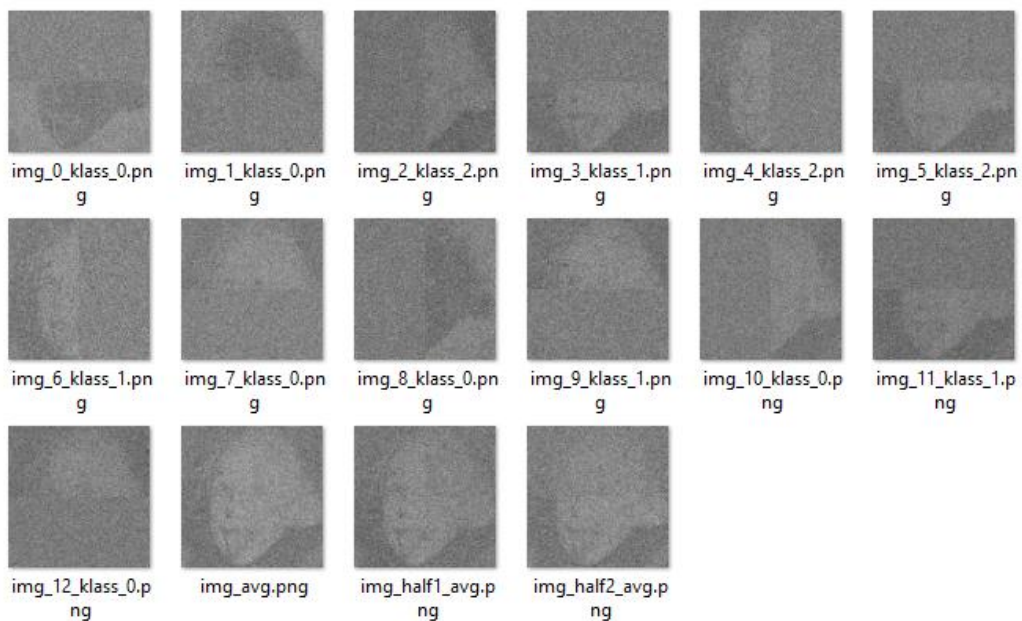
NOTA: La complejidad de los algoritmos se comenta posteriormente en el apartado c) de preguntas.

RESULTADOS

Resultados del algoritmo voraz:

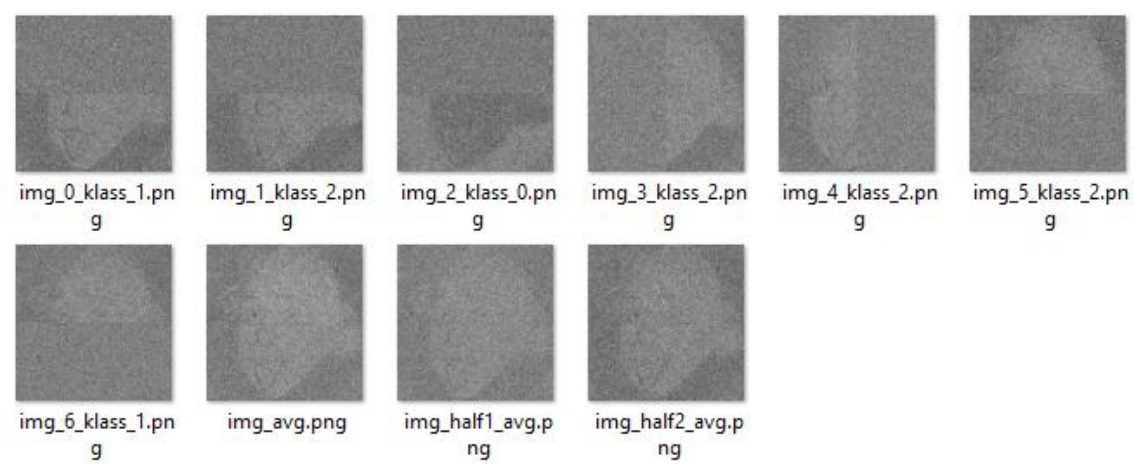
Número de imágenes: 13

TESTING VORAZ:
-ZNCC: 0,069263
-Contador: 13



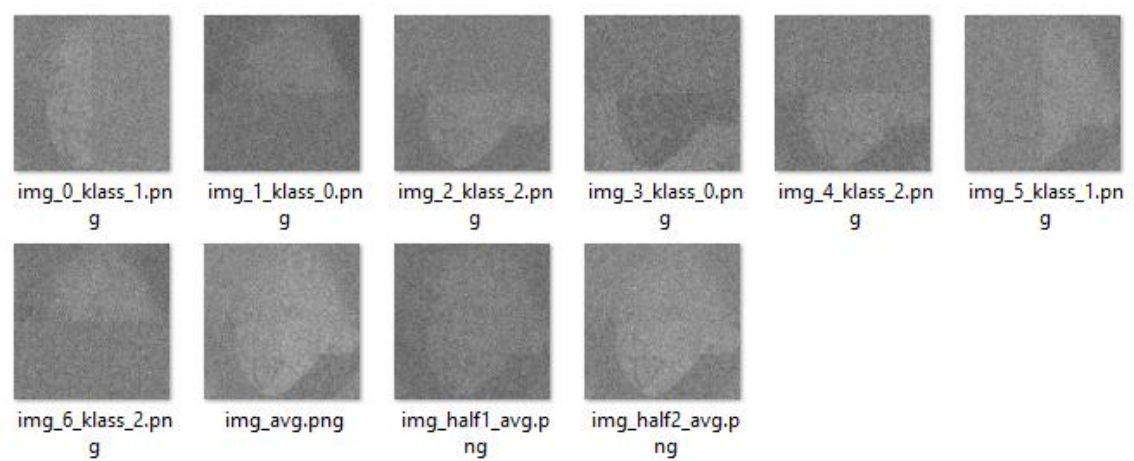
Resultados del algoritmo de backtracking sin balanceo:
Número de imágenes: 7

TESTING BACKTRACKING SIN BALANCEO:
-ZNCC: 0,059400
-Contador: 3279



Resultados del algoritmo de backtracking con balanceo:
Número de imágenes: 7

TESTING BACKTRACKING CON BALANCEO:
-ZNCC: 0,052451
-Contador: 1221



PREGUNTAS

a. ¿Qué algoritmo proporciona mejores resultados y por qué?

La mejor solución la proporciona el algoritmo de backtracking sin balanceo. Esto se debe a que, con este algoritmo, se recorre el árbol de soluciones por completo, y al final se elige la mejor.

En cambio, en el algoritmo de backtracking con balanceo, al haber una condición de poda, no se llega a todas las soluciones, por lo que puede perderse la mejor solución.

b. ¿Qué algoritmo usarías para procesar un conjunto de datos con un millón de imágenes? Explica por qué.

Personalmente, usaría el algoritmo de backtracking con balanceo, ya que es más rápido que el algoritmo sin balanceo, y ofrece resultados casi similares a los de este otro algoritmo.

c. Determina la complejidad temporal del algoritmo backtracking sin considerar la condición de balanceo. Valida este análisis utilizando las medidas experimentales.

Analizando el código del algoritmo, vemos que se hacen 3 llamadas recursivas (una por cada valor posible: 0, 1 o 2). Además, estas llamadas se hacen una vez por cada nivel del árbol de soluciones (la altura de este coincide con el número de imágenes que se utilizan). Por tanto, podemos deducir que la complejidad es $O(3^n)$.

Vamos a validar esto con los datos obtenidos de las mediciones de tiempo, y utilizando la fórmula $t_2 = f(n_2)/f(n_1) \cdot t_1$.

Datos: $t_1=6357\text{ms}$, $n_1=7$, $n_2=8$:

$$t_2 = (3^8)/(3^7) \cdot 6357 = 3 \cdot 6357 = 19071 \text{ ms}$$

Comprobando con el valor obtenido para $n=8$ anteriormente:

$$19071 \text{ ms (valor calculado)} \approx 19906 \text{ ms (valor obtenido)}$$

d. En términos de tiempo, ¿es ventajoso incluir la condición de balanceo? ¿afecta esta condición a la calidad de los resultados?

Sí ofrece una gran ventaja el hecho de añadir la condición de balanceo, ya que el tiempo que tarda en ejecutarse es mucho menor que si no se usa condición de balanceo.

Sí, la condición de balanceo afecta ligeramente al resultado, ya que muchas veces puede que al "podar" una rama, se pierda la solución óptima. Aún así, la diferencia con los resultados del algoritmo sin condición de balanceo no es demasiado grande.