Algorithmics	Student information	Date	Number of session
	UO: 284185	16/4/2022	6
	Surname: Fernández-Catuxo Ortiz		
	Name: Rita		

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Measurements

n	Time_BT	Time_BT_Balancing	ZNCC_greedy	ZNCC_BT	ZNCC_BT_Balancing
2	44	34		0,00426	0,000000
3	42	48		0,025312	0,028179
4	111	119	0,948996	0,015452	0,015804
5	382	281	0,932754	0,018875	0,02302
6	1309	781	0,971292	0,029869	0,029894
7	3427	3008	0,903936	0,037233	0,040954
8	9859	9210	0,843848	0,031772	0,024432
9	31227	34197	0,900034	0,035043	0,031908
10	91253	89677	0,962554	0,040156	0,043465

For testing the zncc in greedy, the first two n numbers (2 and 3) were too small to test this algorithm as they need more images.

I measured up to 10 images for all the algorithms because backtracking with and without balancing with more than 10 images took a lot of time.

Activity 2. Questions

a) State the algorithm that provides better results and explain why.

The algorithm that provides better results is clearly greedy . The table does not show the time measured. However, it is faster (it does not take reliable times until we have more than n = 35 images). For the rest, it takes less than 50 milliseconds.

b) Which algorithm will you use for processing a realistic dataset of a million images? Explain why.

I would use the greedy algorithm because it provides better results, is faster and can deal with more images. (As this is a realistic dataset, we may have a lot of images, and backtracking with or without balancing is a very bad algorithm when dealing with many images).

c) Determine the theoretical time complexity for backtracking (without balancing condition) and validate this analysis from the experimental results.

The complexity of this algorithm is **O(3^n)**, as the degree is 3. We have 3 recursive calls: one for the pictures in the first half, another for the pictures in the second half, and another for the bad pictures.

The number of generated nodes is: O(3^n).

We know that t2 = (f(n2) / f(n1)) x t1. We know that f(n) = 3^n. According to the following results, we can conclude that the values obtained meet the theoretical complexity of the algorithm and thus, they make sense:

Taking this values:

n1 = 5   t1 = 382

n2 = 6   t2 = 1309

t2 = (3^6/3^5) x 382 = 1146 ≈ 1309

Taking this values:

n1 = 8 t1 = 9859

n2 = 9 t2 = 31227

t2 = (3^9/3^8) x 9859 = 29577 ≈ 31227

d) Determine the advantage of including the balancing condition in terms of time for backtracking, does it affect the quality of the results?

As we can see, the results of the backtracking with balancing are better than the backtracking without balancing, and thus the advantage is that it is faster. However, the difference between these results is small.