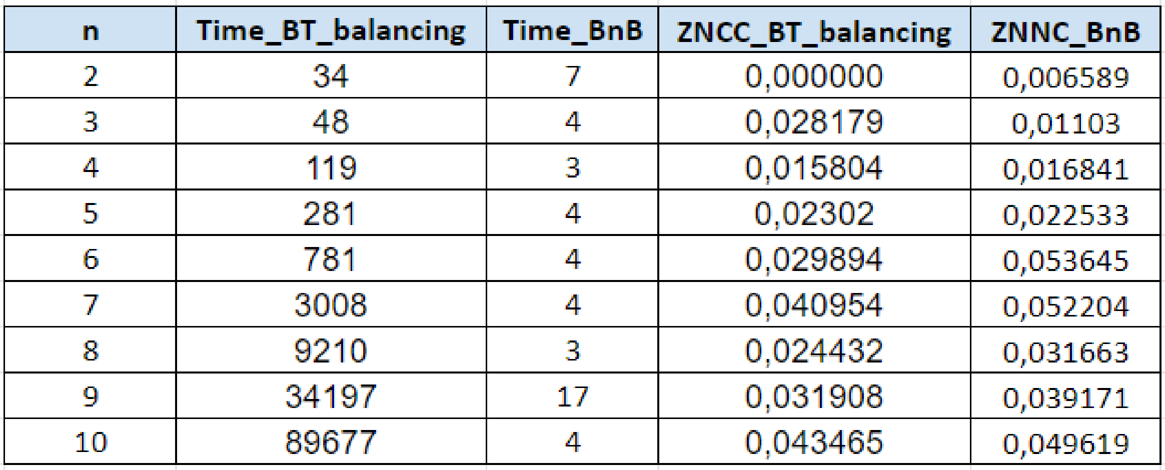
Algorithmics	Student information	Date	Number of session
	UO: 284185	19/4/2022	7
	Surname: Fernández-Catuxo Ortiz		
	Name: Rita		

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Measurements: comparison with Backtracking

1. **Measure times and results for completing the next table**



n	Time_BT_balancing	Time_BnB	ZNCC_BT_balancing	ZNNC_BnB
2	34	7	0,000000	0,006589
3	48	4	0,028179	0,01103
4	119	3	0,015804	0,016841
5	281	4	0,02302	0,022533
6	781	4	0,029894	0,053645
7	3008	4	0,040954	0,052204
8	9210	3	0,024432	0,031663
9	34197	17	0,031908	0,039171
10	89677	4	0,043465	0,049619

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. The time for the branch and bound algorithm is very small, but I couldn't get reliable times with the same nodes for backtracking and branch and bound.

Algorithmics	Student information	Date	Number of session
	UO: 284185	19/4/2022	7
	Surname: Fernández-Catuxo Ortiz		
	Name: Rita		

2. **Compare results, number of nodes, and times for Backtracking (implemented in the previous session) and BnB implementations.**

According to the results, the branch and bound algorithm has better results (it is clearly faster). As with backtracking (depth-first) it is possible to enter a branch with many nodes that do not reach any solution or even that has no end (infinite), this could be a reason to explain why it is slower.

Talking about the nodes, as branch and bound we reach the first solution (not the best one), it uses less nodes than with backtracking, that is going to use all the nodes.

3. **Discuss about the efficiency of both techniques (Backtracking and Brand 'n' Bound) based on the results obtained. Which implementations takes longer times? Which implementation generates more nodes?**

The implementation that takes longer and generates more nodes is backtracking, as branch and bound stops when it reaches the first solution, and backtracking does not stop until in passes through all the nodes.