Algorithmics	Student information	Date	Number of session
	UO: 300829	05/04/2025	6
	Surname: Cid Lazcano		
	Name: Izan		

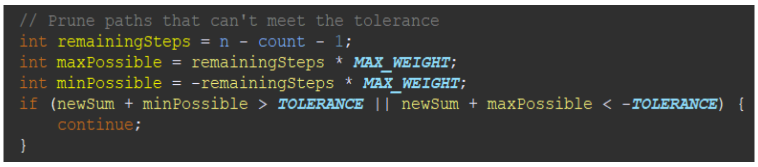
Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Working with graphs

n	Measured Time (ms)
20	0
25	0,03
30	0,1
35	0,1
40	0,2
45	0,31
50	2,7
55	25,74
60	1,08
65	1,86
70	1,8
75	3,81
80	2,86
85	101,83
90	6,69
95	24,23
100	187,88
105	25,91
110	380,26
115	480,4
120	42,92

Although the theoretical time complexity of the algorithm is factorial (meaning it could become extremely slow as the number of nodes increases) in practice, it performs much better thanks to the use of pruning.



```
// Prune paths that can't meet the tolerance
int remainingSteps = n - count - 1;
int maxPossible = remainingSteps * MAX_WEIGHT;
int minPossible = -remainingSteps * MAX_WEIGHT;
if (newSum + minPossible > TOLERANCE || newSum + maxPossible < -TOLERANCE) {
    continue;
}
```

By doing this, the algorithm will discard paths when it is clear that they won't lead to a valid solution, saving a lot of time.

For small graph sizes (n ≤ 45), the algorithm runs almost instantly, showing that pruning is very effective.

Between n = 50 and 85, the execution times begin to vary more depending on the random structure of each graph. Some cases are solved quickly, while others take longer, but the times still remain reasonable.

From n = 90, we observe some spikes in execution time, such as 187 ms, 380 ms, or even 480 ms. These are expected in graphs where pruning can’t eliminate as many paths. Even so, the times do not grow nearly as fast as they would in a purely factorial algorithm.

If the algorithm truly followed factorial growth, increasing n just from 20 to 25 would result in a great time increase (which doesn’t happen here).