Activity 1. Backtracking with graphs

|  |  |
| --- | --- |
| **n** | **t Backtracking (ms)** |
| 20 | 0.080467 |
| 25 | 0.178919 |
| 30 | 0.32252 |
| 35 | 0.41224 |
| 40 | 0.899015 |
| 45 | 2.014633 |
| 50 | 9.117259 |
| 55 | 2.94463 |
| 60 | 47.619499 |
| 65 | 11.25175 |
| 70 | 17.012901 |
| 75 | 110.366254 |
| 80 | 162.678025 |

The theoretical complexity of the **NullPath** algorithm is **O(n!)**, since in the worst case, the backtracking explores all possible permutations of paths. However, several factors significantly reduce the actual growth in practical scenarios:

* Random Graph Generation (Luck Factor):
  + As we generate several random graphs in some of them, there is no easy path, so we need to check more paths in order to get the exact solution.
* Pruning Heuristic:
  + The pruning step reduces the number of branches explored by eliminating paths that **cannot** reach a valid solution within the given constraints.

Finally, this **NullPath algorithm** is O(n!) but improved, to reduce the time in each iteration.

And with it the average case.