TSP - TRAVELLING SALESMAN PROBLEM

n = 2
real   0m0.012s
user   0m0.004s
sys   0m0.000s

n = 3
real   0m0.009s
user   0m0.000s
sys   0m0.004s

n = 4
real   0m0.012s
user   0m0.005s
sys   0m0.000s

n = 5
real   0m0.008s
user   0m0.003s
sys   0m0.000s

n = 6
real   0m0.009s
user   0m0.004s
sys   0m0.000s

n = 7
real   0m0.008s
user   0m0.003s
sys   0m0.001s

n = 8
real   0m0.011s
user   0m0.004s
sys   0m0.000s

n = 9
real   0m0.019s
user   0m0.013s
sys   0m0.002s

n = 10
real   0m0.077s
user   0m0.070s
sys   0m0.002s

n = 11
real   0m0.621s
user   0m0.605s
sys   0m0.012s

n = 12
real   0m7.433s
user   0m7.426s
sys   0m0.002s

n = 13
real   1m37.792s
user   1m37.786s
sys   0m0.002s

n = 14
real   23m5.614s
user   23m5.585s
sys   0m0.021s

Como o algoritmo possui complexidade de $O(n!)$, o tempo cresce muito mesmo para n's pequenos. O número máximo de n que o algoritmo encontrou solução com um tempo razoável foi n = 14.

OP - ORIENTEERING PROBLEM

n = 1; T = 0
real   0m0.009s
user   0m0.002s
sys   0m0.001s

n = 1; T = 10000
real   0m0.009s
user   0m0.002s
sys   0m0.000s

n = 2; T = 0
real   0m0.011s
user   0m0.002s
sys   0m0.000s

n = 2; T = 10000
real   0m0.010s
user   0m0.002s
sys   0m0.000s

n = 3; T = 0
real   0m0.008s
user   0m0.002s
sys   0m0.000s

n = 3; T = 10000
real   0m0.008s
user   0m0.002s
sys   0m0.000s

n = 4; T = 0
real   0m0.008s
user   0m0.002s
sys   0m0.001s

n = 4; T = 10000
real   0m0.010s
user   0m0.003s
sys   0m0.001s

```
n = 5; T = 0
real    0m0.007s
user    0m0.002s
sys     0m0.000s


n = 5; T = 10000
real    0m0.010s
user    0m0.002s
sys     0m0.000s


n = 6; T = 0
real    0m0.008s
user    0m0.002s
sys     0m0.000s


n = 6; T = 10000
real    0m0.009s
user    0m0.003s
sys     0m0.000s


n = 7; T = 0
real    0m0.011s
user    0m0.005s
sys     0m0.000s


n = 7; T = 10000
real    0m0.010s
user    0m0.005s
sys     0m0.000s


n = 8; T = 0
real    0m0.029s
user    0m0.022s
sys     0m0.000s


n = 8; T = 10000
real    0m0.031s
user    0m0.024s
sys     0m0.000s


n = 9; T = 0
real    0m0.222s
user    0m0.214s
sys     0m0.000s
```

n = 9; T = 10000
real   0m0.210s
user   0m0.203s
sys   0m0.000s

n = 10; T = 0
real   0m2.093s
user   0m2.086s
sys   0m0.000s

n = 10; T = 10000
real   0m2.077s
user   0m2.071s
sys   0m0.000s

n = 11; T = 0
real   0m24.479s
user   0m24.462s
sys   0m0.011s

n = 11; T = 10000
real   0m24.367s
user   0m24.350s
sys   0m0.010s

n = 12; T = 0
real   5m27.175s
user   5m27.169s
sys   0m0.001s

n = 12; T = 10000
real   5m31.465s
user   5m31.459s
sys   0m0.001s

Como o algoritmo precisa gerar todos os subconjuntos e permutar cada subconjunto, o tempo cresce muito rápido mesmo para n's muito pequenos. Além disso, o valor de T não influencia no tempo total, pois sempre é necessário gerar todos os subconjuntos e permutar cada subconjunto para descobrir a solução ótima. E o algoritmo encontra uma solução em um tempo razoável com um n de no máximo 12.