Relatório 1º Projeto

Neste relatório foi pedido que para certos problemas fossem analisados e comparados os resultados obtidos com uma procura em profundidade primeiro, uma procura gananciosa e uma procura A*. Para isso, precisamos de resolver o jogo Same Game utilizando cada uma destas procuras. Após a execução deste problema analisaremos o tempo de execução, o número de nós expandidos e o número de nós gerados para cada uma as procuras e faremos uma comparação em relação aos resultados obtidos, à completude, à eficiência e à heurística. Para obtermos os resultados necessários iremos utilizar a classe presente no ficheiro *search.py* chamada "InstrumentedProblem".
Nos testes realizados sobre estas procuras utilizámos 4 tabuleiros aleatórios de tamanhos diferentes todos com solução. O primeiro é um tabuleiro 3X4, o segundo 8X4, o terceiro 12X4 e o último 15X4. A diferença entre tamanhos dos tabuleiros é necessária para podermos observar melhor qual das procuras é a mais eficiente e se existe uma correlação entre a execução do programa e o tamanho dos tabuleiros.

Procura em Profundidade Primeiro

A procura em profundidade primeiro consiste em expandir o nó na fronteira com maior profundidade. Os testes com esta procura obtiveram os seguintes resultados:
—>1º tabuleiro:
-Tempo de execução: 0.001232 segundos
-Número de nós gerados: 6
-Número de nós expandidos: 5
—>2º tabuleiro:
-Tempo de execução: 0.003318 segundos
-Número de nós gerados: 8
-Número de nós expandidos: 7
—>3º tabuleiro:
-Tempo de execução: 0.38661 segundos
-Número de nós gerados: 1338
-Número de nós expandidos: 1339
—>4º tabuleiro:
-Tempo de execução: 0.03616 segundos
-Número de nós gerados: 16
-Número de nós expandidos: 17

Procura Gananciosa

A procura gananciosa consiste na utilização de uma função de objetivo, função heurística, que é uma estimativa do caminho com menos custo desde o nó *n* até ao objetivo. Este tipo de procura expande o nó que "parece" estar mais próximo do objetivo. Relativamente aos resultados dos testes efetuados:
—>1º tabuleiro:
-Tempo de execução: 0.001344 segundos
-Número de nós gerados: 7
-Número de nós expandidos: 5

—>2º tabuleiro:
-Tempo de execução: 0.003461 segundos
-Número de nós gerados: 6
-Número de nós expandidos: 4
—>3º tabuleiro:
-Tempo de execução: 0.009193 segundos
-Número de nós gerados: 9
-Número de nós expandidos: 7
—>4º tabuleiro:
-Tempo de execução: 0.024155 segundos
-Número de nós gerados: 14
-Número de nós expandidos: 16

Procura A*

A procura A* tem como ideia principal evitar expandir caminhos que já têm um custo muito elevado. Contem também uma função avaliação que é igual à soma da função do custo do nó inicial até ao nó *n* com a função heurística que é uma estimativa do custo desde o nó *n* até um estado objetivo. Podemos dizer então que a função objetivo é a estimativa do custo da melhor solução que passa por um determinado nó *n* até uma solução. Depois de realizar os testes, obtivemos os seguintes resultados:
—>1º tabuleiro:
-Tempo de execução: 0.002846 segundos
-Número de nós gerados: 10
-Número de nós expandidos: 8
—>2º tabuleiro:
-Tempo de execução: 0.008077 segundos
-Número de nós gerados: 7
-Número de nós expandidos: 5
—>3º tabuleiro:
-Tempo de execução: 0.045221 segundos
-Número de nós gerados: 20
-Número de nós expandidos: 18
—>4º tabuleiro:
-Tempo de execução: 0.074279 segundos
-Número de nós gerados: 17
-Número de nós expandidos: 15

Conclusão

Após a análise dos resultados obtidos para cada procura concluímos que a procura menos eficiente é a procura em profundidade primeiro apesar de no 1º tabuleiro ser a que expande menos nós até à solução isto apenas se deve ao facto de o tabuleiro ser de dimensões pequenas e por isso ser fácil encontrar a solução. Pelos resultados obtidos em cima podemos ver que existe uma relação entre o tamanho do tabuleiro e o tempo de execução para qualquer uma das procuras, ou seja, quanto maior é o tabuleiro maior o tempo de execução. Podemos dizer então que todas as procuras são completas pois todas chegam a uma solução.