**Trabalho 1 Inteligência Artificial**

**Problema 1 (g1)**

Espaço de Estados e operadores de transição

Para a representação do problema apresentado foi escolhido o estado (linha, coluna), onde é representado a posição do agente A. No caso apresentado a representação do estado inicial é (7,2), e a saída seria (1,5), para os diferentes cálculos foi antes utilizada a saída (3,4) devido a alguns problemas de memoria (global stack overflow).

Uma imagem com texto

Descrição gerada automaticamente

**Análise dos Algoritmos**

Os algoritmos utilizados tanto na pesquisa informada como não informada foram retirados dos algoritmos disponibilizados durante as aulas.

**Pesquisa não informada**

Para executar este tipo de pesquisas basta utilizar o ficheiro “pni.pl” com o comando “pesquisa(<problema>, <algoritmo>)”, sendo os algoritmos possíveis: “it” (profundidade iterativa), “largura”, “profundidade”.

Durante a análise dos diferentes algoritmos foi apenas possível retirar conclusões relativamente ao algoritmo de largura e profundidade iterativa, devido a problemas relacionados com global stack overflow, não foi possível retirar qualquer conclusão relativamente á pesquisa em profundidade.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algoritmo | Estados Visitados | Estados em Memória | Profundidade | Custo |
| Largura | 813 | 1799 | 6 | 6 |
| Profundidade Iterativa | 45103 | 13 | 6 | 6 |

Comparando a pesquisa em largura com profundidade iterativa podemos concluir que ambas encontram a solução á mesma profundidade e custo. No entanto embora a pesquisa em largura tenha menos estados visitados, tem mais estados em memória.

**Pesquisa Informada**

Para executar este tipo de pesquisas basta utilizar o ficheiro “pi.pl” com o comando “pesquisa(<problema>, <algoritmo>)”, sendo os algoritmos possíveis: “a” (A\*) ou “g” (Gready). É também possível alterar o tipo de heurística no final do ficheiro relativo ao problema, bastando alterar a heurística (“h”) comentada.

Heurísticas

Para representar duas heurísticas admissíveis para o problema em questão foi utilizada a distância de Manhattan e a distância euclidiana.

Uma imagem com texto

Descrição gerada automaticamente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algoritmo | Estados Visitados | Estados em Memória | Profundidade | Custo |
| A\*, euclidiana | 408 | 324 | 6 | 6 |
| A\*, Manhattan | 210 | 236 | 6 | 6 |
| Gready, euclidiana | 10 | 27 | 6 | 6 |
| Gready, Manhattan | 145 | 215 | 8 | 8 |

Analisando todos os algoritmos, concluímos que o melhor algoritmo é o gready com a heurística calculada pela distância euclidiana.

**Problema 2 (g2)**

Espaço de Estados e operadores de transição

Para a representação do problema apresentado foi escolhido o estado (linha jogador, coluna jogador, linha caixa, coluna caixa), onde é representado a posição do agente A. No caso apresentado a representação do estado inicial é ( (7, 2), (6, 2) ), e a saída seria ((\_,\_),(1,5)), sendo que a posição do jogador não importa desde que a caixa se encontre no ponto pretendido. para os diferentes cálculos foi antes utilizada a saída((\_,\_),(6,3)), devido a alguns problemas de memoria (global stack overflow).

Uma imagem com texto

Descrição gerada automaticamente

**Análise dos Algoritmos**

Pesquisa não informada

Durante a análise dos diferentes algoritmos foi apenas possível retirar conclusões relativamente ao algoritmo de largura e profundidade iterativa, devido a problemas relacionados com global stack overflow, não foi possível retirar qualquer conclusão relativamente á pesquisa em profundidade.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algoritmo | Estados Visitados | Estados em Memória | Profundidade | Custo |
| Largura | 17 | 36 | 3 | 3 |
| Profundidade Iterativa | 185 | 6 | 3 | 3 |

Comparando a pesquisa em largura com profundidade iterativa podemos concluir que ambas encontram a solução á mesma profundidade e custo. No entanto embora a pesquisa em largura tenha menos estados visitados, tem mais estados em memória.

**Pesquisa Informada**

Heurísticas

Para representar duas heurísticas admissíveis para o problema em questão foi utilizada a distância de Manhattan e a distância euclidiana.

Uma imagem com texto

Descrição gerada automaticamente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algoritmo | Estados Visitados | Estados em Memória | Profundidade | Custo |
| A\*, euclidiana | 11 | 15 | 3 | 3 |
| A\*, Manhattan | 8 | 11 | 3 | 3 |
| Gready, euclidiana | 6 | 8 | 3 | 3 |
| Gready, Manhattan | 5 | 5 | 3 | 3 |

Analisando todos os algoritmos, concluímos que o melhor algoritmo é o gready com a heurística calculada pela distância de Manhattan.