

Considere o puzzle Sokoban (<http://en.wikipedia.org/wiki/Sokoban>), que consiste em mover as caixas “\$” para as casas com as marcas finais “.”. O trabalhador que tem a marca “@”, pode mover-se pelos espaços livres na horizontal e vertical, empurrando uma caixa, mas não conseguindo empurrar duas caixas. Pode mover-se também pelos espaços que têm a marca final “.”, ficando nesse caso com a representação de “+”. As caixas têm a marca “\$” se não estiverem na marca final, ficando “*” nas posições de marca final. O trabalhador não pode saltar caixas, nem passar por paredes, que têm a marca “#”.

Exemplo de um problema:

```
#####
##.@ #
# $$ #
#. ##
#####
Author: Jordi Domenech
Title: Level 1
Limit: 10
```

O problema acima começa numa linha com apenas parede, e termina com uma linha também com apenas parede. Os textos posteriores são informação relativa ao puzzle.

O problema poderia ser resolvido pela seguinte sequência de movimentos válidos:

<pre>##### ##.@ # # \$\$ # #. ## ##### ##### ##+ # # \$\$ # #. ## ##### ##### ##. # # @\$ # #.\$ ## ##### ##### ##+ # # \$ # #.\$ ## #####</pre>	<pre>##### ##.@ # # \$ # #.\$ ## ##### ##### ##. @# # \$ # #.\$ ## ##### ##### ##. # # @\$# #.\$ ## ##### ##### ##. # # @\$ # #.\$ ## #####</pre>	<pre>##### ##. # # \$ # #.\$@## ##### ##### ##. # # \$ # #* @ ## ##### ##### ##* # # @ # #* ## #####</pre>
---	--	--

Alternativamente, pode-se mostrar apenas movimentos que empurrem caixas, desde que exista sempre possibilidade de se expandir esta solução para a solução incluindo todos os movimentos:

<pre>##### ##.@ # # \$\$ # #. ## #####</pre>	<pre>##### ##. # # @\$ # #.\$ ## #####</pre>	<pre>##### ##. # # @\$ # #.\$ ## #####</pre>
--	--	--

```
#####
##. #
# $ #
#* @ ##
#####
```

```
#####
##* #
# @ #
#* ##
#####
```

Os testes podem ser realizados com base no ficheiro **Pufiban.txt**, de Jordi Domenech, que contém problemas de dimensão reduzida neste formato, no entanto algumas com soluções bastante longas. Pretende-se que possa ser carregado qualquer um dos problemas, devendo o programa aceitar os seguintes argumentos:

C:\...>Sokoban <ficheiro> <problema>

O número do problema é 1 para o primeiro problema, 2 para o segundo, e assim sucessivamente. No caso do ficheiro fornecido, existem 170 problemas, todos com um número reduzido de caixotes, sendo este o tipo de problema que se pretende resolver com apenas técnicas de procuras cegas.

Deve entregar:

- Relatório;
- Código fonte dos algoritmos implementados.

Critérios de correção (4 valores):

- **Análise do problema** (2 valores): Referência a aspectos importantes do problema no relatório, revelando independentemente de os implementar ou não, que tinha consciência dos mesmos.
- **Identificação de algoritmos** (1 valor): Identificação clara dos algoritmos que implementou de acordo com a nomenclatura do livro e da UC, juntamente com as configurações utilizadas, ou no caso de utilização de um algoritmo distinto, deve descrevê-lo. A utilização de outro nome para os mesmos algoritmos é possível, desde que indique a qual correspondente. A penalização para a não identificação corresponde a 0,5 valores.
- **Resultados** (1 valor): Deve procurar resolver pelo menos um problema, e caso consiga resolver vários, deve procurar resolver o problema de nível mais elevado (deve sempre utilizar menos de 1 minuto de CPU, modo de *release*), e indicar o algoritmo/configuração, comprimento da solução, número de expansões e gerações.

O trabalho é individual mas caso os estudantes pretendam, podem partilhar resultados. A partilha de resultados afasta o problema de uma situação real, em que não existindo referências, não se sabe até onde se consegue chegar, mas pode contribuir para uma maior participação no e-fólio, e em nada afeta a avaliação. Os resultados obtidos através da resolução de exemplo, serão conhecidos após o lançamento das notas.