

## 1 Ćwiczenie 3

1. Zapewne pamiętasz o odkurzaczu omówionym w pierwszych dwóch wykładach.
  - (i) Sformułuj problem jako problem przeszukiwania; tzn. musisz napisać stan początkowy, możliwe działania za pomocą funkcji **Action(state)**, relację przejścia za pomocą funkcji **Result(state, action)** i stan celu. (Zobacz strona 7 wykładu 2). Zakładamy, że oba kwadraty A i B są zabrudzone.
  - (ii) Napisz program za pomocą algorytm 'Breadth First Search, aby agent mógł odszukiwać brud we wszystkich możliwych lokalizacjach i wyczyścić go. Wystarczy, aby program wydrukował każdą sekwencję akcji na konsoli, i pokazał kiedy osiąga cel.
2. Rozważ 8 zagadek ze stanem początkowym jako  $\begin{bmatrix} 8 \\ 6 \\ 3 \end{bmatrix} \begin{bmatrix} 7 \\ 5 \\ 2 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix}$ .
  - (i) Opisz sformułowanie problemu jako problem przeszukiwania.
  - (ii) Napisz program, używając 'greedy first search'. Możesz użyć funkcji heurystycznej, gdzie  $h(\text{stan}) = \text{suma odległości bloku miasta (sum of city block distance)}$ . Wystarczy, aby program wydrukował każdą sekwencję akcji na konsoli, i pokazał kiedy osiąga cel.
3. Zapewne pamiętasz o Trywialnej grze rozważanej w wykładach 3 i 4 (Zobacz strona 5 wykładu 4). Napisz program, aby znaleźć optymalną strategię dla MAX za pomocą algorytmu Maximin z przycinaniem  $[\alpha, \beta]$  ( $[\alpha, \beta]$ -pruning).