

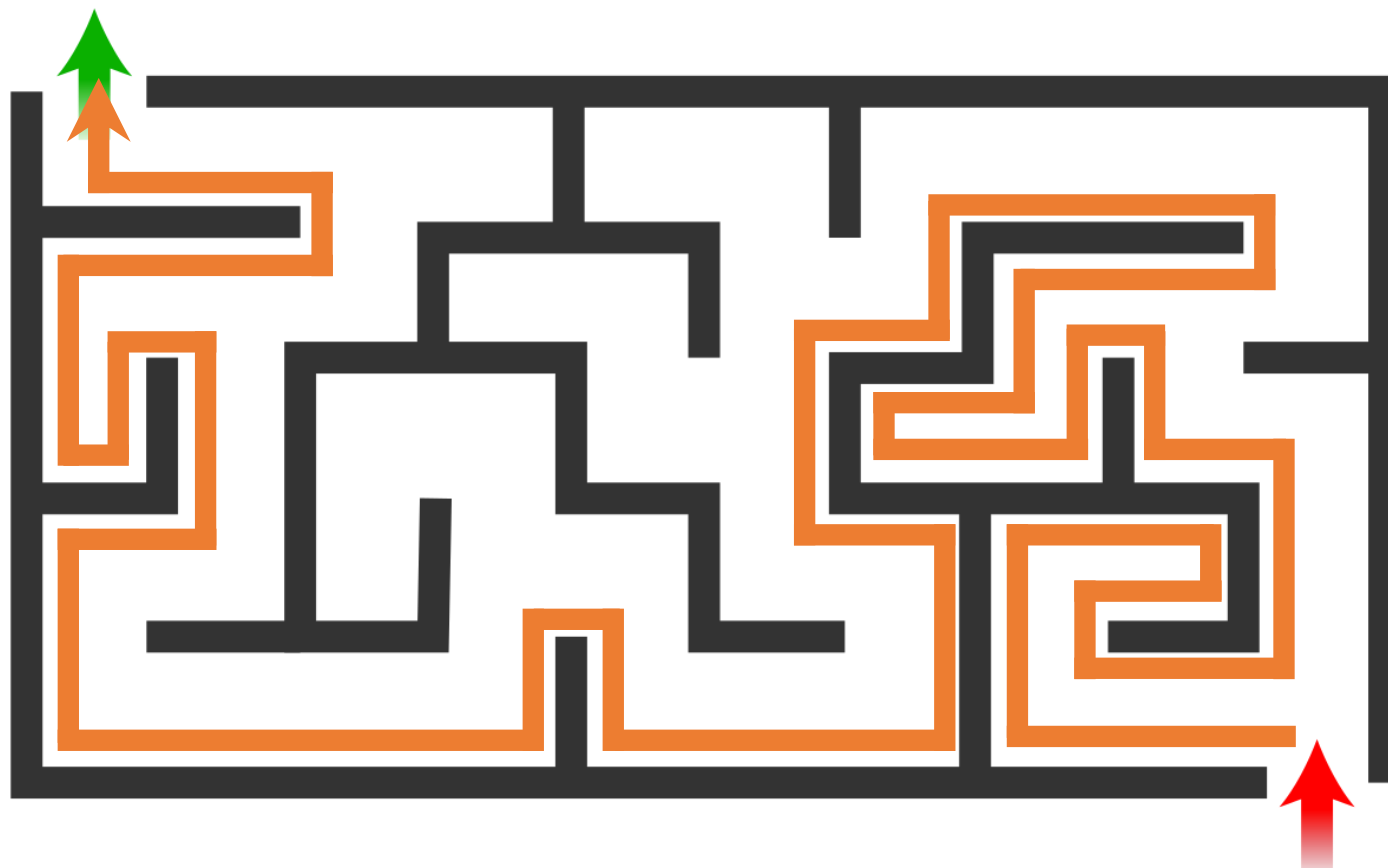
# Metoda lewej ręki

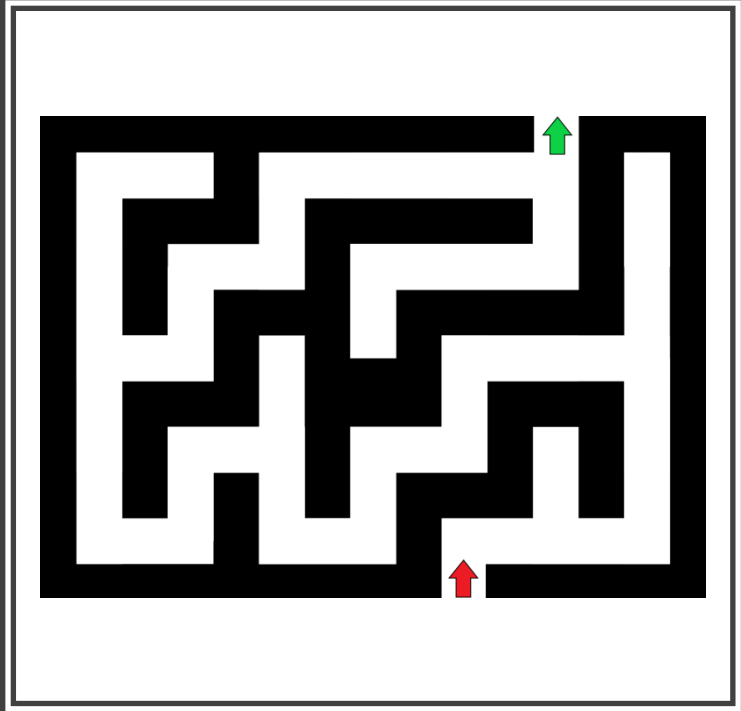
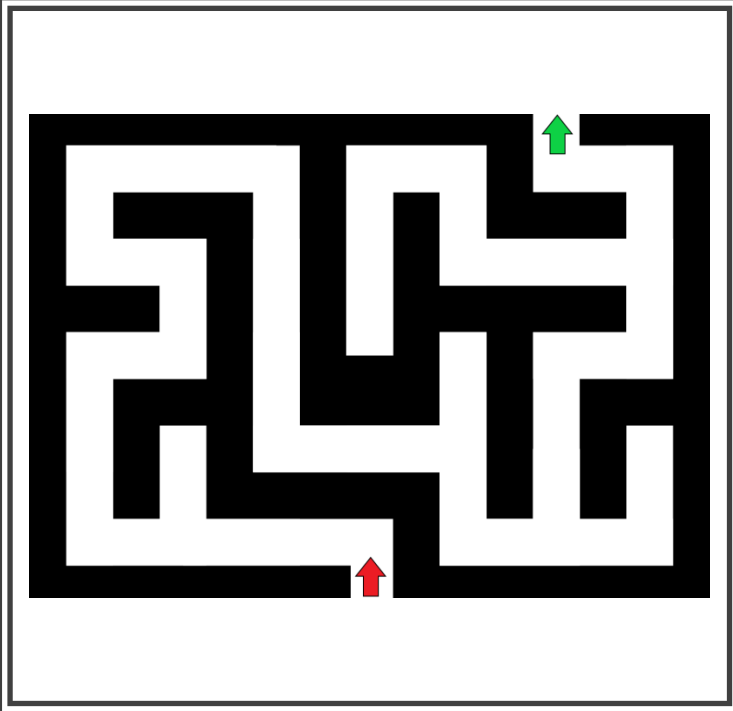
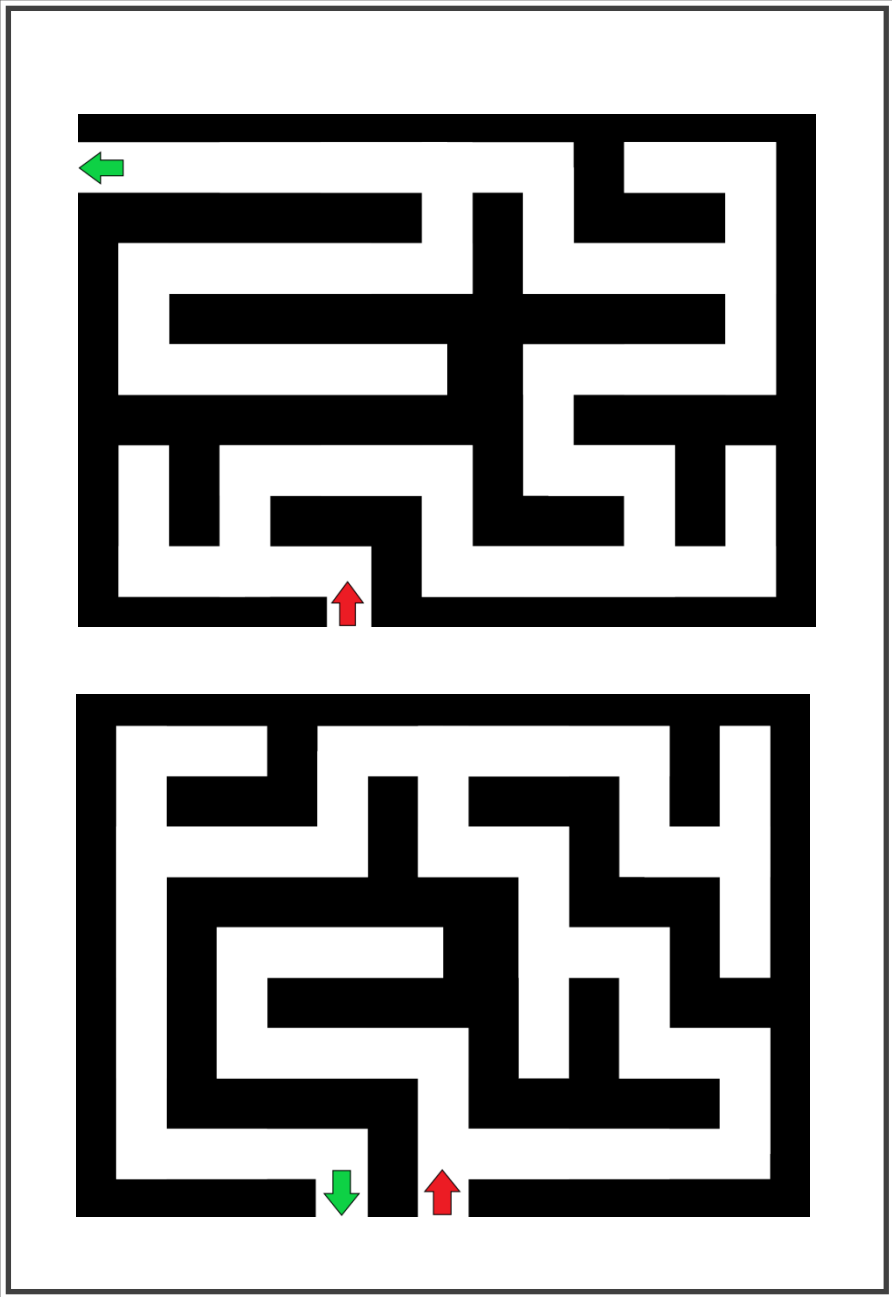
Czyli jak wyjść z labiryntu





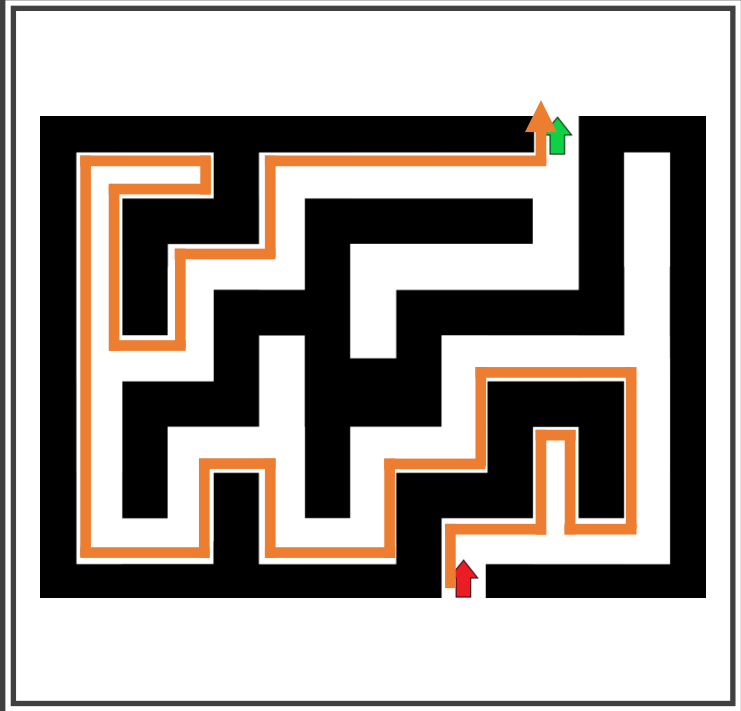
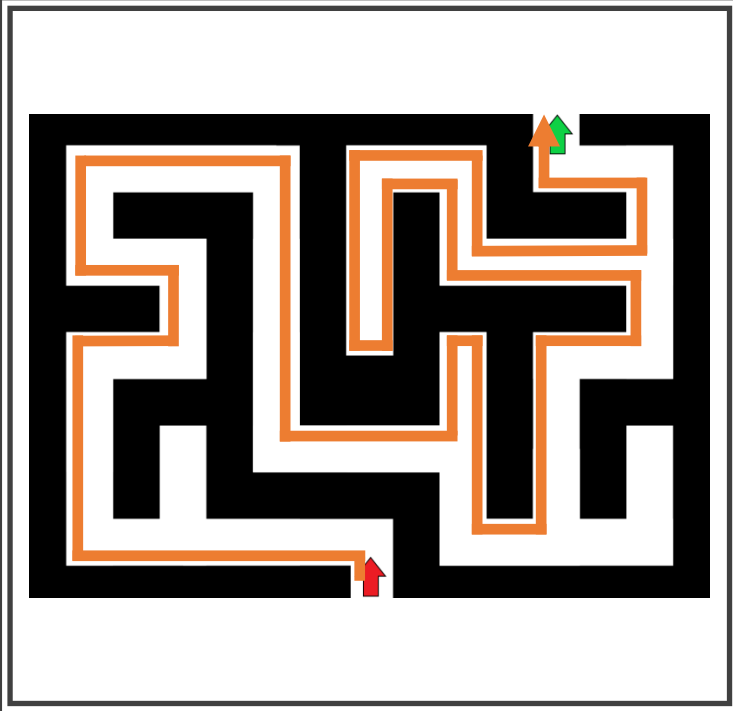
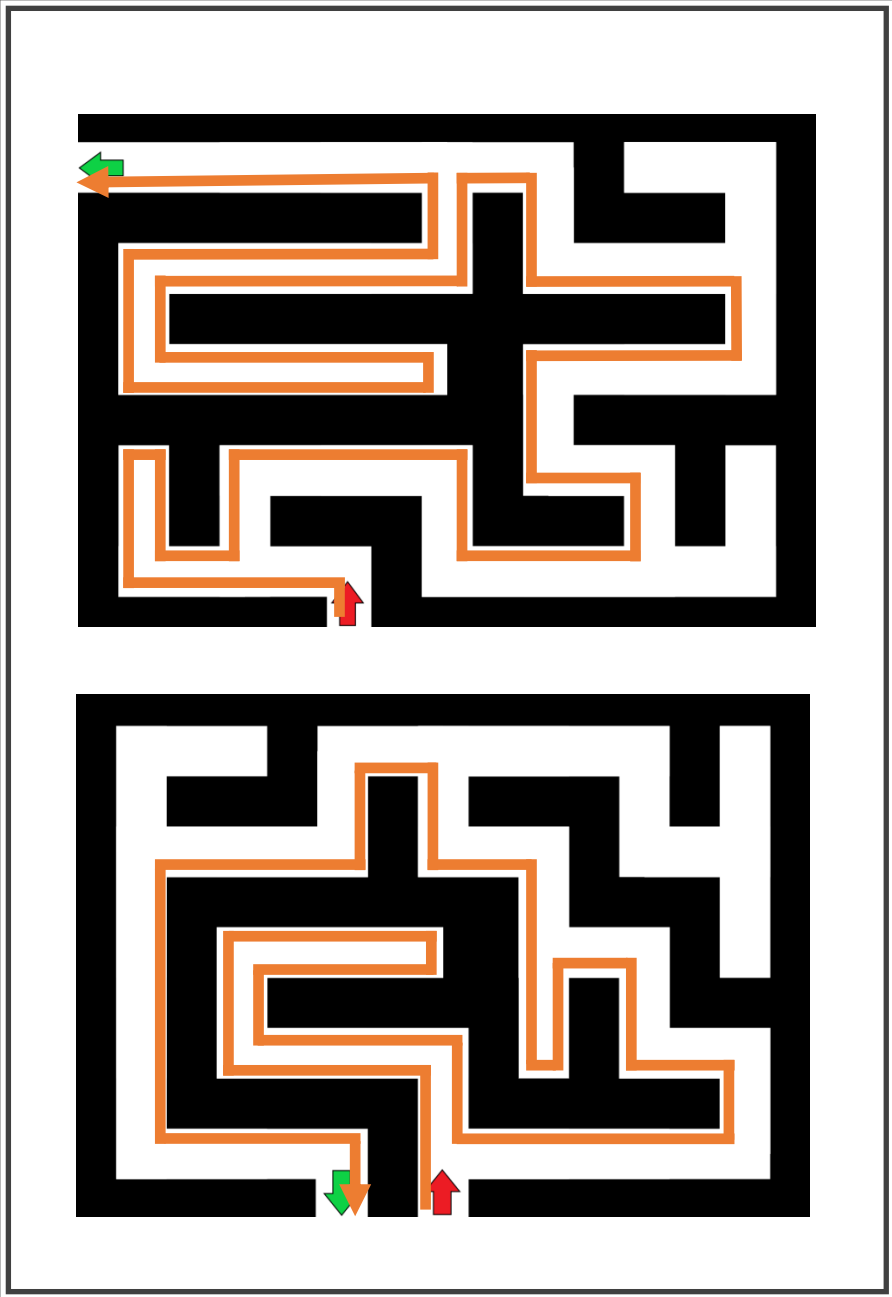
Działanie w  
praktyce





Zadanie – zastosować metodę lewej ręki na podanym labiryncie

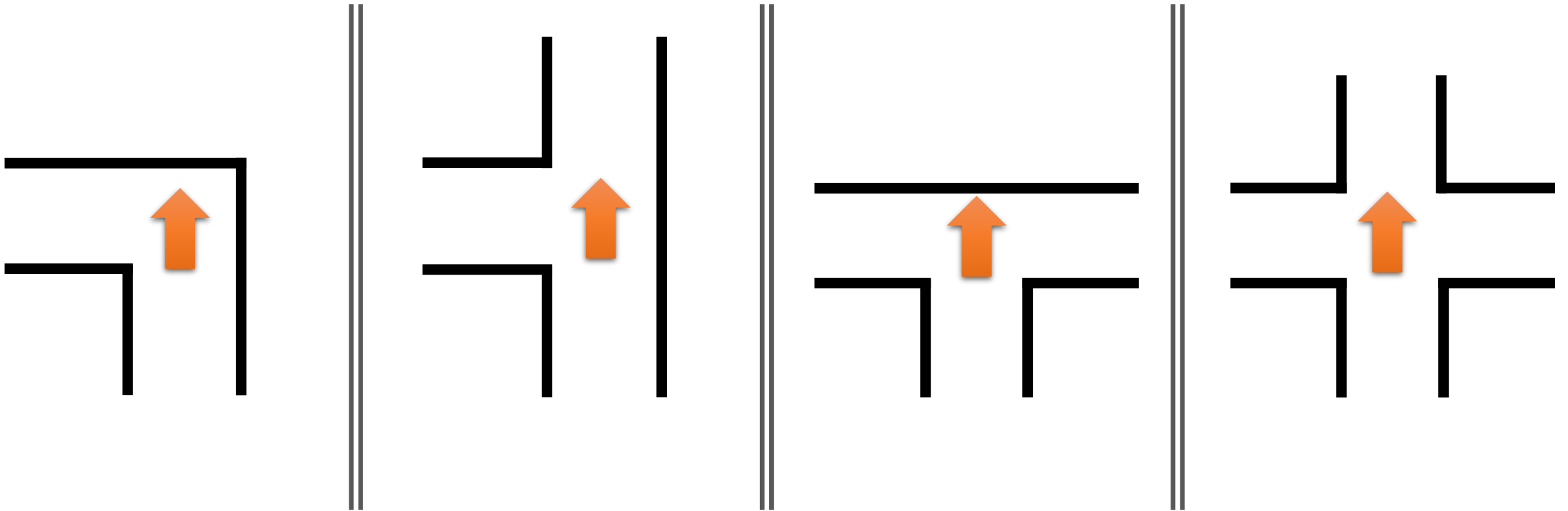
---



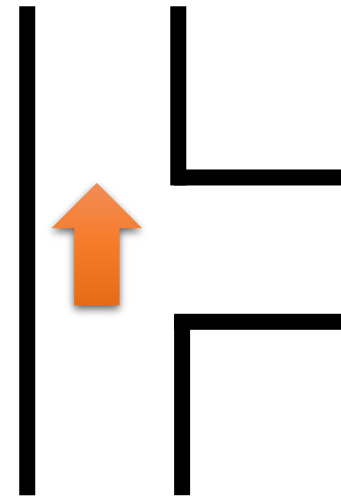
Rozwiązanie zadania

---

Idąc labiryntem,  
możemy napotkać kilka  
odmiennych sytuacji



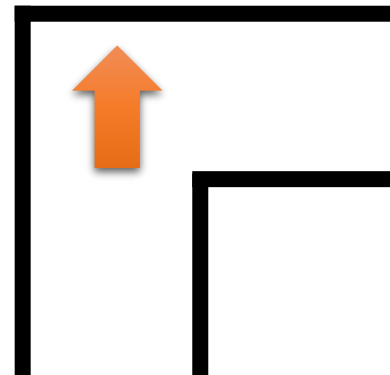
Korytarz prowadzi w lewo



Korytarz prowadzi prosto, ale nie w lewo

---





Korytarz nie prowadzi ani w lewo, ani prosto

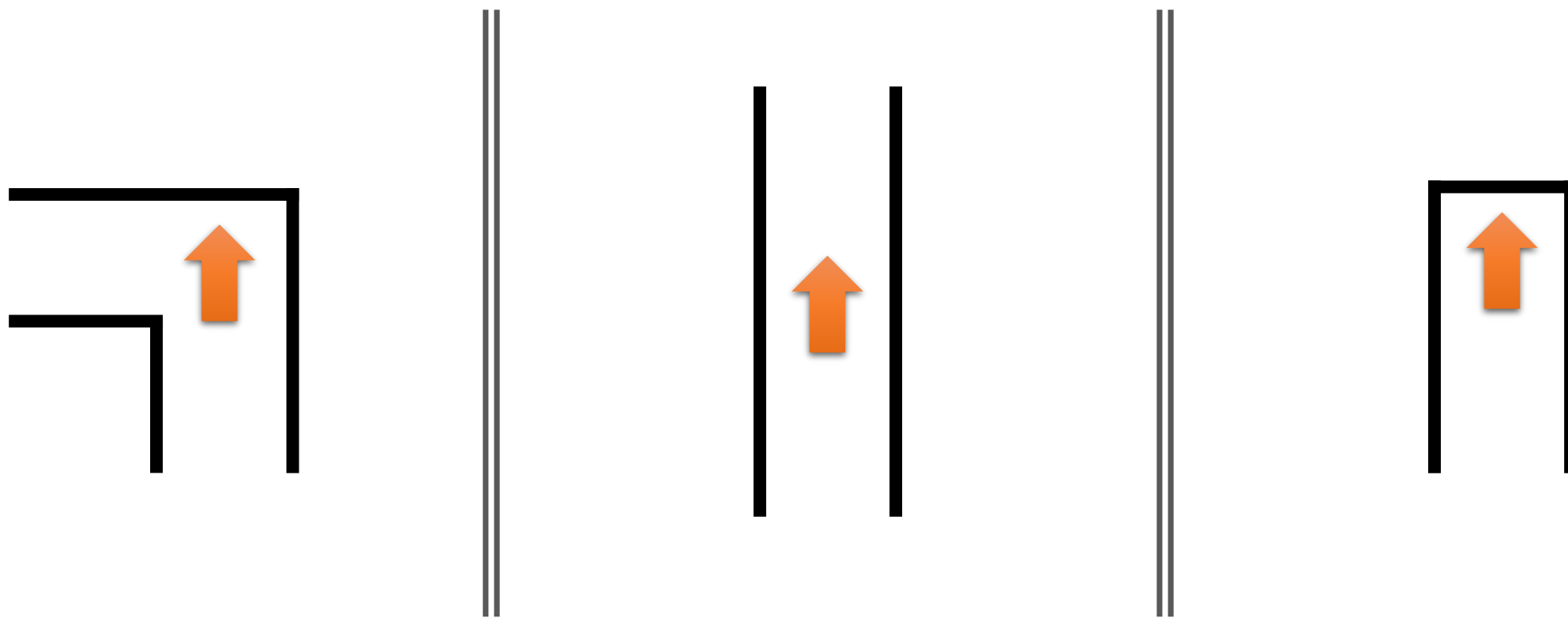
---

W każdej z tych sytuacji  
musimy inaczej się  
zachować

W ten sposób  
konstruujemy nasz  
algorytm –  
wyjścia z labiryntu

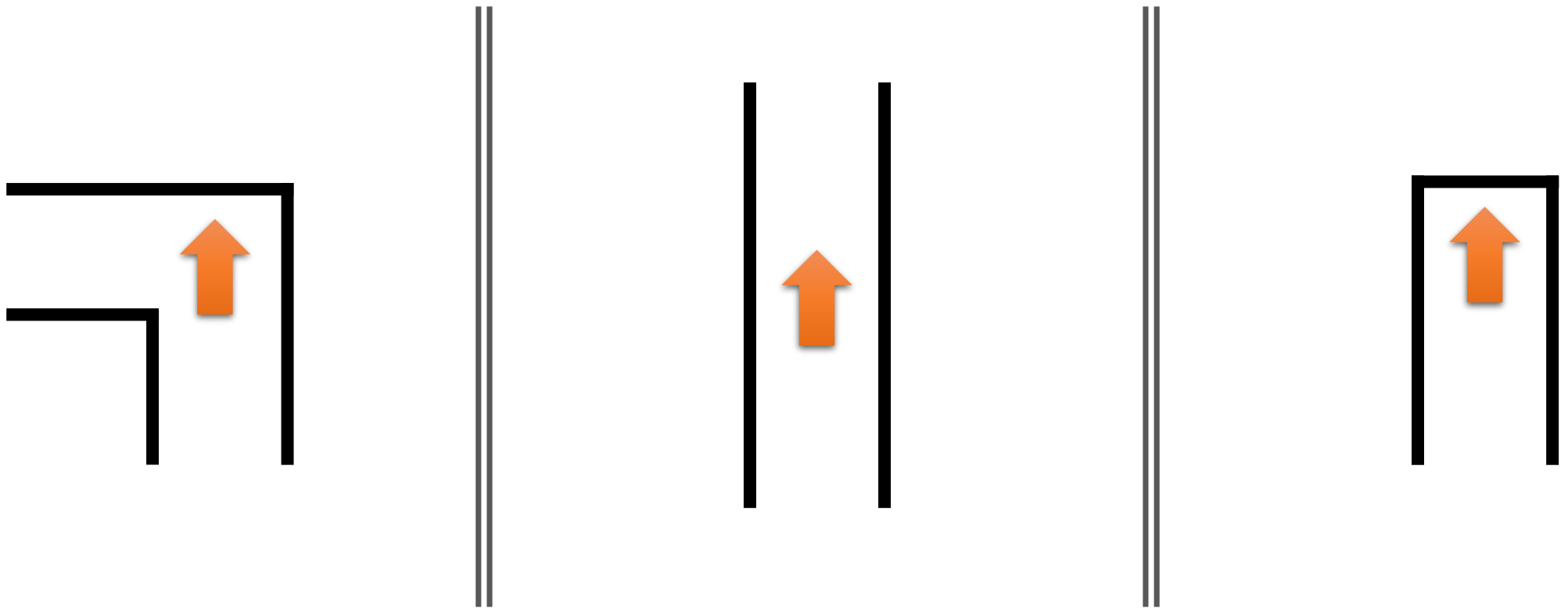
W każdej sytuacji  
wykonujemy określone  
akcje.

One doprowadzą nas  
do nowej sytuacji.



Możliwe sytuacje – jak się w nich zachować?

---

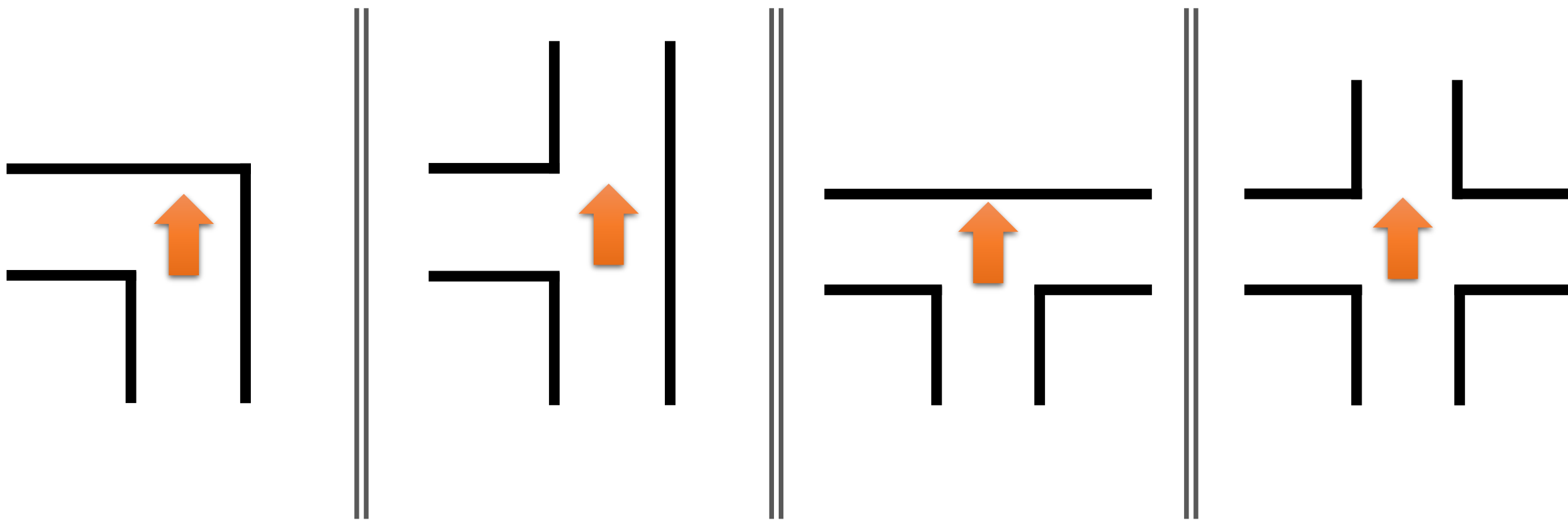


Inaczej – jak je zaprogramować?

---

# Zasady ogólne

- Musimy pamiętać, aby **zawsze** trzymać się ściany po **naszej lewej stronie**
- Mamy do dyspozycji następujące akcje: **skręt o 90° w lewo/prawo** oraz **krok naprzód**
- Możemy wykonać **jedną** lub **więcej** akcji, ale coś **musimy** zrobić
- Wykonanie tych akcji doprowadzi nas znowu do **jednej z trzech sytuacji...**
- Ale przecież będziemy już wtedy **wiedzieli, co należy zrobić!**

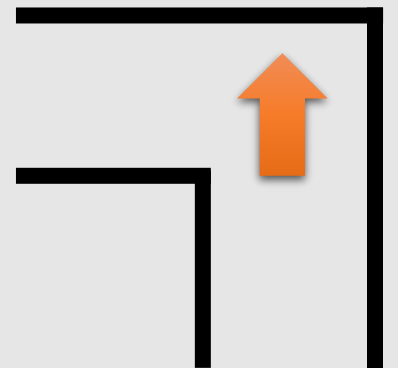


Sytuacja 1. Korytarz prowadzi w lewo



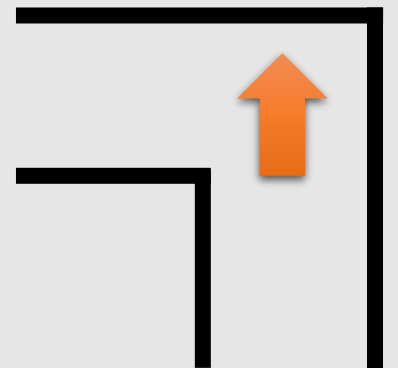
# Sytuacja 1. Korytarz prowadzi w lewo

- Musimy skręcić w **lewo**, tak prowadzi nas ściana...



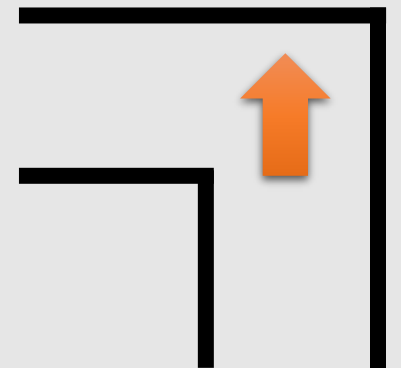
# Sytuacja 1. Korytarz prowadzi w lewo

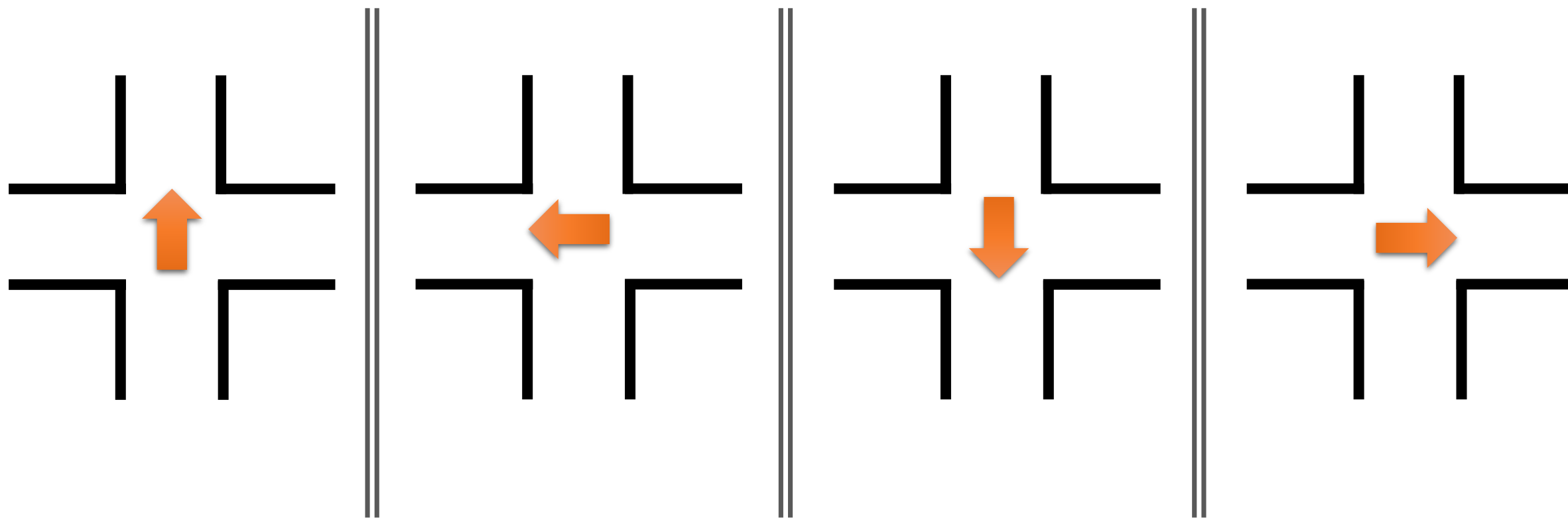
- Musimy skręcić w **lewo**, tak prowadzi nas ściana...
- Ale skręt w lewo **nie wystarczy!** Dlaczego?



# Sytuacja 1. Korytarz prowadzi w lewo

- Musimy skręcić w **lewo**, tak prowadzi nas ściana...
- Ale skręt w lewo **nie wystarczy!** Dlaczego?
- Zauważ, że ograniczenie się do skrętu w lewo będzie skutkowało oderwaniem lewej ręki od ściany (w dalszych ruchach)



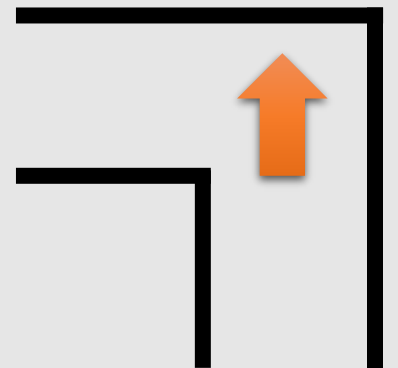


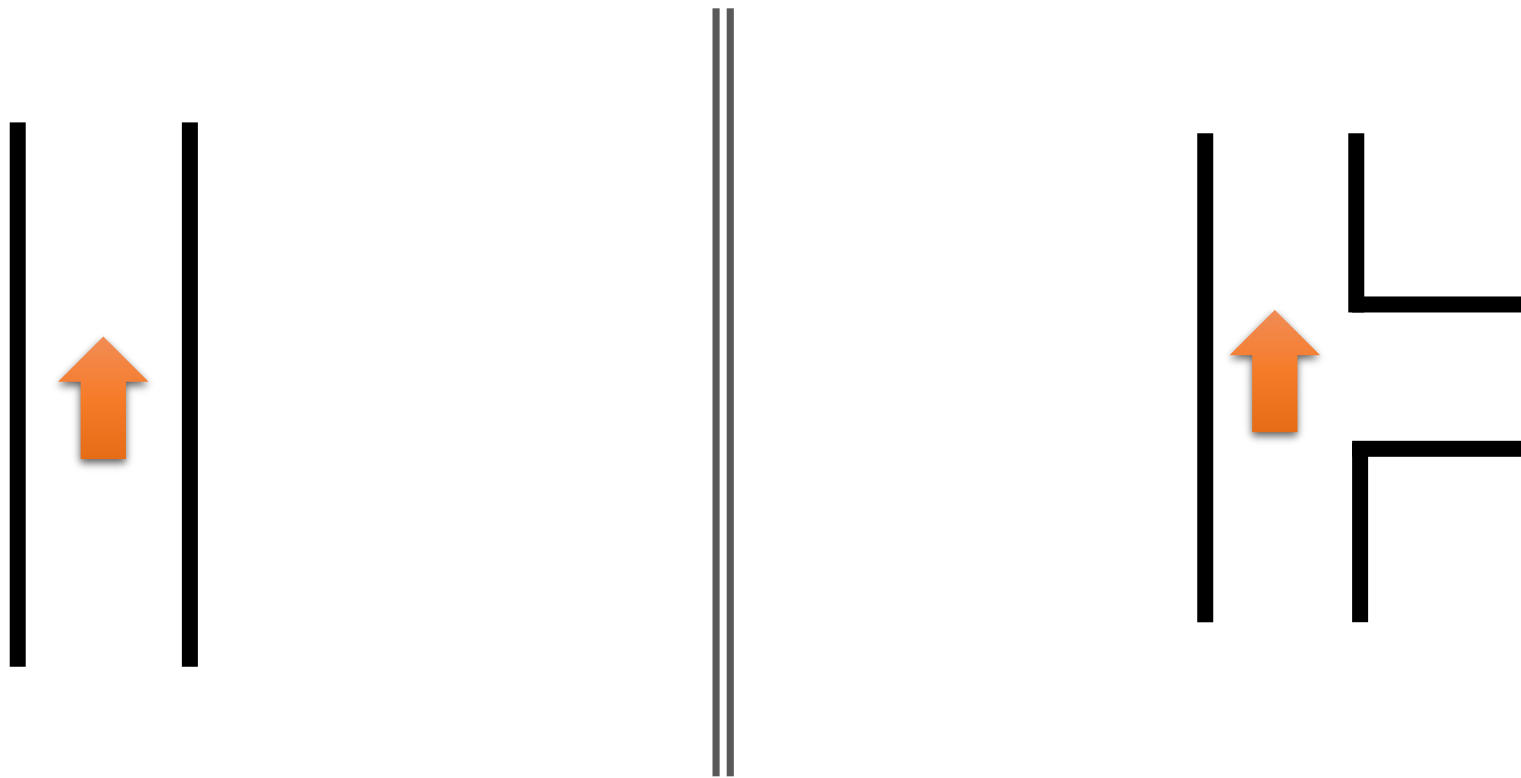
Sytuacja 1. Wykonanie tylko obrotu w lewo

---

# Sytuacja 1. Korytarz prowadzi w lewo

- Musimy skręcić w **lewo**, tak prowadzi nas ściana...
- Ale skręt w lewo **nie wystarczy!** Dlaczego?
- Skoro korytarz prowadzi w lewo, to możemy nim podążać
- **Skręcamy w lewo i robimy krok naprzód**





Sytuacja 2. Korytarz prowadzi prosto, ale nie w lewo

---

## Sytuacja 2. Korytarz prowadzi prosto, ale nie w lewo

- Możemy iść prosto, ściana po lewej stronie nie zakręca...

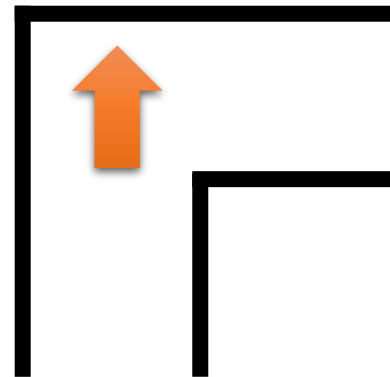
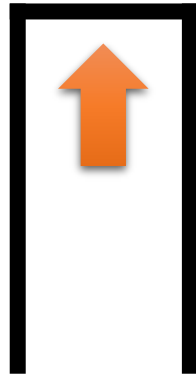


## Sytuacja 2. Korytarz prowadzi prosto, ale nie w lewo

- Możemy iść prosto, ściana po lewej stronie nie zakręca...
- Więc robimy **krok naprzód!**





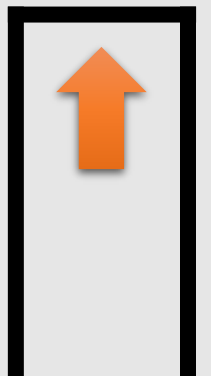


Sytuacja 3. Korytarz nie prowadzi ani w lewo, ani prosto

---

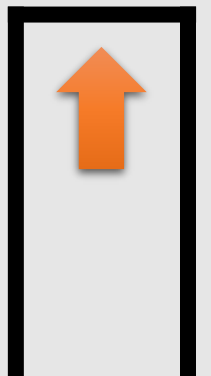
Sytuacja 3. Korytarz nie prowadzi ani w lewo, ani prosto

- Nie możemy iść prosto ani skręcić w lewo
- **Oznacza to, że lewa ściana zakręca przed nami**



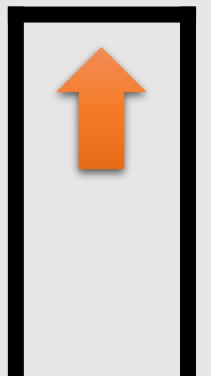
## Sytuacja 3. Korytarz nie prowadzi ani w lewo, ani prosto

- Nie możemy iść prosto ani skręcić w lewo
- Oznacza to, że lewa ściana zakręca przed nami
- **Trzymamy się lewej ściany i skręcamy razem z nią**



## Sytuacja 3. Korytarz nie prowadzi ani w lewo, ani prosto

- Nie możemy iść prosto ani skręcić w lewo
- Oznacza to, że lewa ściana zakręca przed nami
- Trzymamy się lewej ściany i skręcamy razem z nią
- **Wykonujemy skręt w prawo**



# Konstrukcja rozwiązania - algorytmu

- I. Badamy w której z trzech wyróżnionych sytuacji się znajdujemy
  1. Jeżeli w pierwszej (**jest droga w lewo**) – wykonujemy **skręt w lewo i krok naprzód**
  2. Jeżeli w drugiej (**jest droga naprzód, nie ma w lewo**) – wykonujemy **krok naprzód**
  3. Jeżeli w trzeciej (**nie ma drogi w lewo ani naprzód**) – wykonujemy **skręt w prawo**
- II. Powtarzamy punkt I. aż do znalezienia wyjścia z labiryntu

# Zwróć uwagę na metodę postępowania przy tworzeniu algorytmu

- Uwzględniamy **wszystkie możliwe** sytuacje
- **Grupujemy** je w oparciu o **konieczną reakcję**
- **Dla każdej** z grup określamy **niezbędne działania**
- Projektujemy sposób określania **do której grupy** należy sytuacja, w której się znajdujemy
- Projektujemy sposób powtarzania działań aż do otrzymania **wyniku – znalezienia wyjścia**

# Zwróć uwagę na specyficzny sposób postępowania

1. Mamy dane (labirynt) – analizujemy je.
2. Znamy oczekiwany wynik działań, które podejmujemy – znalezienie wyjścia.  
**Punkty 1. i 2. tworzą tzw. Specyfikację zadania**
3. Zastanawiamy się nad możliwymi sposobami rozwiązania. Podejmujemy decyzję - będziemy posuwać się labiryntem trzymając się lewą ręką ściany tak długo, aż trafimy na wyjście.
4. Z obserwacji wnioskujemy o wszystkich możliwych sytuacjach w trakcie posuwania się po labiryncie i koniecznych w każdej z tych sytuacji działań.
5. Grupujemy sytuacje ze względu na konieczne działania.
6. Projektujemy mechanizm stwierdzania w jakiej sytuacji jesteśmy, wyboru i wykonania odpowiednich działań.
7. Zapisujemy utworzony **ALGORYTM** w postaci listy kroków powtarzanych aż do znalezienia wyjścia. Algorytm możemy zapisać w postaci ciągu instrukcji w języku programowania.

- Opisany wyżej sposób postępowania jest wynikiem pewnego specyficznego sposobu myślenia (i działania), zwanego **myśleniem komputacyjnym** lub - wężiej - **myśleniem algorytmicznym**.
- Najpierw, opierając się na intuicji i doświadczeniu, wybieramy ogólną metodę postępowania – tak powstaje ogólny zarys algorytmu.
- Myślenie komputacyjne zakłada m.in. **redukcję i dekompozycję** złożonych problemów. Tak postąpiliśmy: (redukcja labiryntu do możliwych sytuacji i koniecznych zachowań) i to umożliwiło uzyskanie precyzyjnego rozwiązania – **algorytmu**.



# Rozwiązanie alternatywne

- I. Badamy w której z dwóch sytuacji się znajdujemy
  1. Jeżeli **jest droga w lewo** – wykonujemy **skręt w lewo i krok naprzód**
  2. W przeciwnym wypadku (**nie ma drogi w lewo**) – wykonujemy **skręt w prawo**
- II. Powtarzamy punkt I. aż do znalezienia wyjścia z labiryntu

# Źródła

- <https://pixabay.com/en/maze-labyrinth-solution-lost-1804499/>
- [https://commons.wikimedia.org/wiki/File:Maze\\_simple.svg](https://commons.wikimedia.org/wiki/File:Maze_simple.svg)