# Mrówka Langtona

## Opis algorytmu

Algorytm mrówki Langtona opiera się na śledzeniu, jak zachowuje się określona plansza po i iteracjach. Mrówka chodząc po planszy skręca w lewo gdy będzie na polu białym oraz skręca w prawo gdy natrafi na pole czarne. Poruszając się w ten sposób, jednocześnie zmieniając kolor pola, znajdującego się pod nią, algorytm rysuje chaotyczny, jednak deterministyczny rysunek, aby następnie stworzyć odnogę, którą mrówka będzie się poruszała do końca iteracji programu, bądź gdy natrafi na przeszkodę.

## Wywołanie

Program jest uruchamiany z pliku wykonywalnego, stworzonego przez kompilator c. Plik ten nazwany jest „mrowka" i znajduje się on w folderze bin. Można go zaktualizować wywołując plik „Makefile", znajdujący się w folderze głównym programu.

Plik wykonywalny jesteśmy w stanie uruchomić za pomocą polecenia „./bin/mrowka -n [n] – m [m] -i [i] -d [dir] -p [%]", gdzie dane wejściowe odpowiadają: n – liczba wierszy, m – liczba kolumn, i – liczba iteracji, dir – kierunek początkowy mrówki [od 0 do 3], % - procent zapełnienia planszy czarnymi polami.

## Podział modułowy

Program jest podzielony na 3 pliki z rozwinięciem „.c" oraz 2 pliki z rozwinięciem „.h".

Plik „main.c" zawiera funkcję „main" , która odpowiada za sczytanie danych wejściowych oraz wywołanie programu.

Plik „ant.c" odpowiada za tworzenie, destrukcję(zwolnienie z pamięci) oraz ruch mrówki. Odpowiadają za to odpowiednio funkcje: „create_ant", „destroy_ant", „step". Powiązanym z tym plikiem jest plik „ant.h" zawierający prototypy funkcji zdefiniowanych w „ant.c" oraz definicje stałych, które będzie używał ten program.

Plik „board.c" odpowiada za tworzenie, destrukcję(zwolnienie z pamięci) oraz wypisanie planszy po której będzie poruszać się mrówka. Za te działania odpowiadają odpowiednio funkcje: „create_board", „destroy_board", „save_board". Powiązanym plikiem jest „board.h". Zawiera on prototypy funkcji zdefiniowanych w „board.c"

W całym programie znajduje się jedna własna struktura nazwana „Ant". Zawiera ona 3 zmienne int. Oznaczają one odpowiednio: wiersz, kolumnę oraz kierunek; mrówki.

## Przykłady

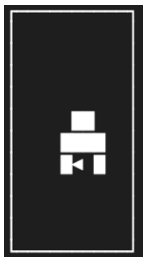
1.

Dla wywołania „./bin/mrowka -n 10 -m 10 -i 20 -d 0 -p 0 -f wyj”, stworzy się 20 plików w folderze „out”

```
MaciejK@MacBook-Pro-4 out % ls
wyj_1.txt       wyj_12.txt      wyj_15.txt      wyj_18.txt      wyj_20.txt      wyj_5.txt       wyj_8.txt
wyj_10.txt      wyj_13.txt      wyj_16.txt      wyj_19.txt      wyj_3.txt       wyj_6.txt       wyj_9.txt
wyj_11.txt      wyj_14.txt      wyj_17.txt      wyj_2.txt       wyj_4.txt       wyj_7.txt
```

Te pliki odpowiadają odpowiednim iteracjom przejścia mrówki i zawierają plansze wypełnione białymi oraz czarnymi polami. Zaznaczona jest też w nich pozycja mrówki.

Przykładowy wygląd pliku(wyj_16.txt):



*

2.

Gdy program zostanie wywołany z innymi danymi „20 30 30 1 50”, można zauważyć, że tworzy się 30 plików tak jak zakłada program.

```
MaciejK@MacBook-Pro-4 out % ls
test_1.txt      test_17.txt     test_24.txt     test_4.txt      wyj_11.txt      wyj_19.txt      wyj_8.txt
test_10.txt     test_18.txt     test_25.txt     test_5.txt      wyj_12.txt      wyj_2.txt       wyj_9.txt
test_11.txt     test_19.txt     test_26.txt     test_6.txt      wyj_13.txt      wyj_20.txt
test_12.txt     test_2.txt      test_27.txt     test_7.txt      wyj_14.txt      wyj_3.txt
test_13.txt     test_20.txt     test_28.txt     test_8.txt      wyj_15.txt      wyj_4.txt
test_14.txt     test_21.txt     test_29.txt     test_9.txt      wyj_16.txt      wyj_5.txt
test_15.txt     test_22.txt     test_3.txt      wyj_1.txt       wyj_17.txt      wyj_6.txt
test_16.txt     test_23.txt     test_30.txt     wyj_10.txt      wyj_18.txt      wyj_7.txt
```

Każdy z plików wygenerowany przez ten program będzie miał 20 wierszy oraz 30 kolumn.

Pierwsza iteracja czyli plik nazwany „test_1.txt” zawiera już czarne pola, co jest definiowane przez ostatni argument, tworzący planszę początkową z 50% czarnymi polami. Sama mrówka obrócona jest w prawo co definiuje 2 argument.

Przykładowy wygląd pliku(test_1.txt):

*

** - czarne kwadraty są zamienione na białe i na odwrót. Jest to spowodowane ciemnym motywem edytora tekstu*

## Wnioski

Algorytm Mrówki Langtona obrazuje jak chaotyczne ruchy mrówki zmieniają się w uporządkowane, po ok. 10000 iteracji. Jest to o tyle ciekawe, że jest to algorytm deterministyczny, czyli przy każdym wywołaniu, będzie wyglądał tak samo(nie wliczając 5 argumentu, który tworzy planszę losową).

Mimo ilości możliwych iteracji jakie może wykonać program wykonuje się on relatywnie szybko dla stosunkowo małej liczby wierszy i kolumn. Algorytm ten wykonuje się w złożoności n*m*i.