

Zaawansowane algorytmy — 3-kolorowanie cyklu

Mateusz Stapaj

22 czerwca 2024

1 Wstęp

Celem projektu jest stworzenie aplikacji, która będzie wykorzystywała wcześniej przygotowane algorytmy do rozwiązania problemu 3-kolorowania cyklu oraz problemu odwrotnego 3-kolorowania cyklu. Do rozwiązania tego problemu stworzyłem aplikację w języku programowania Java. Dostępny jest interfejs, który umożliwia wygodne skorzystanie ze stworzonych algorytmów. Dodatkowo zostały przeprowadzone testy wydajności przygotowanych algorytmów w zależności od podanych danych.

2 Opis problemów

2.1 3-kolorowanie cyklu

Problem 3-kolorowania cyklu polega na określeniu kolorów dla wierzchołków w cyklu w taki sposób, aby kolory sąsiednich wierzchołków były inne od koloru sprawdzanego wierzchołka. Dane wejściowe problemu to cykl opisany za pomocą tablicy z liczbami, które oznaczają identyfikatory procesorów (początkowe kolory cyklu). Wyjściem problemu jest tablica kolorów, których ilość została zredukowana do maksymalnie trzech różnych kolorów.

2.2 Odwrotne 3-kolorowanie cyklu

Problem odwrotnego 3-kolorowania cyklu polega na znalezieniu początkowego kolorowania dla cyklu o określonym rozmiarze, wcześniej ustalonej ilości iteracji algorytmu 3-kolorowania cyklu, z lub bez uwzględnienia usuwania kolorów 5, 4, 3. Dane wejściowe problemu to wielkość cyklu, oczekiwana ilość iteracji algorytmu, czy algorytm ma wykonać usuwanie kolorów 5, 4, 3. Wyjściem problemu jest tablica z początkowym kolorowaniem wierzchołków cyklu.

3 Przygotowane algorytmy

3.1 Colorize - 3-kolorowanie cyklu

Do rozwiązania problemu 3-kolorowania cyklu wykorzystałem podstawowy algorytm 3-kolorowania cyklu.

W tym rozwiązaniu wykorzystywana jest procedura BasicColoring, która zmienia kolory wierzchołków, dopóki wszystkie kolory nie będą mniejsze niż 6. Procedura ta zamienia wartości kolorów na liczby binarne. Następnie dla każdego wierzchołka wyznacza najmniej znaczący bit, który różni się od bitu na tej samej pozycji w następnym wierzchołku w cyklu. Indeks bitu jest przypisywany do zmiennej k . Wartość sprawdzanego wierzchołka na miejscu różniącego się indeksu jest przypisywana do zmiennej bit . Na koniec obliczany jest nowy kolor za pomocą wzoru $c=2k+bit$.

Jeżeli wszystkie kolory wierzchołków są mniejsze niż 6, to wtedy usuwane są następująco liczby 5, 4, 3. Dla każdego wierzchołka, którego wartość wynosi więcej niż 2, zmieniana jest wartość na najmniejszą wartość (większą lub równą 0) różniącą się od sąsiednich kolorów.

Na koniec zwracana jest tabela z ostatecznymi kolorami wierzchołków, które zawierają kolory o numerach 0, 1, 2.

3.2 Reverse Colorize — odwrotne 3-kolorowanie cyklu

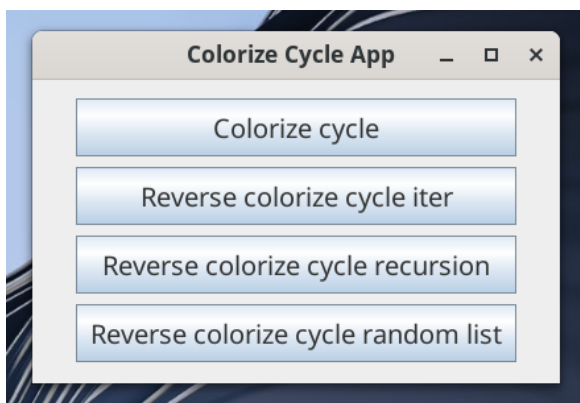
Do rozwiązania problemu odwrotnego 3-kolorowania cyklu została wykorzystana metoda "brute force". Zostały utworzone trzy algorytmy, które generują listę potencjalnych rozwiązań tego problemu:

- generowanie wszystkich permutacji liczb o ustalonym rozmiarze i maksymalnej wartości koloru — algorytm rekurencyjny,
- generowanie wszystkich permutacji liczb o ustalonym rozmiarze i maksymalnej wartości koloru — algorytm iteracyjny,
- generowanie list z losowymi liczbami o ustalonym rozmiarze i maksymalnej wartości koloru.

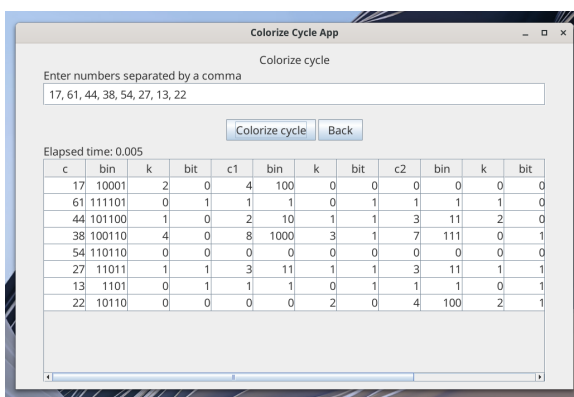
Następnie przy pomocy algorytmu Colorize sprawdzane jest, czy jedno z potencjalnych rozwiązań spełnia wymagania rozwiązania.

4 Wygląd aplikacji

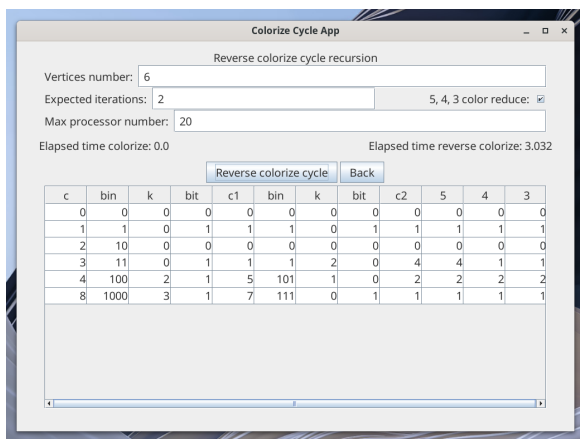
Aplikacja zawiera interfejs umożliwiający skorzystanie z algorytmu 3-kolorowania cyklu lub z algorytmów odwrotnego 3-kolorowania cyklu. W programie jest dostępne menu, które umożliwia wybór algorytmu. Po wybraniu jednej z opcji wyświetlany jest widok, który umożliwia skorzystanie z wybranego algorytmu. Przed uruchomieniem obliczeń należy podać wymagane dane. W przypadku błędnych danych lub błędu podczas obliczeń pojawi się komunikat z odpowiednią informacją.



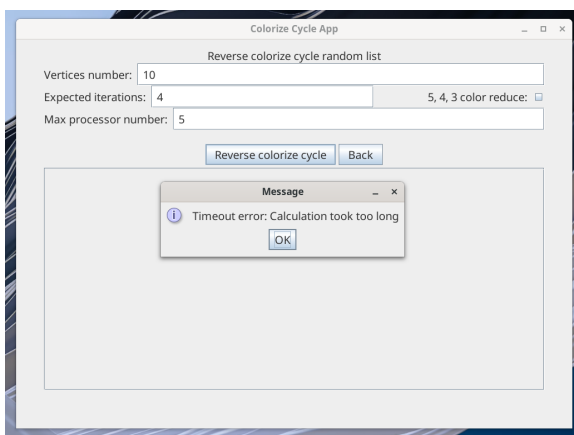
(a) Menu główne aplikacji z dostępnymi algorytmami



(b) Widok algorytmu kolorowania cyklu



(c) Widok algorytmu odwrotnego kolorowania cyklu



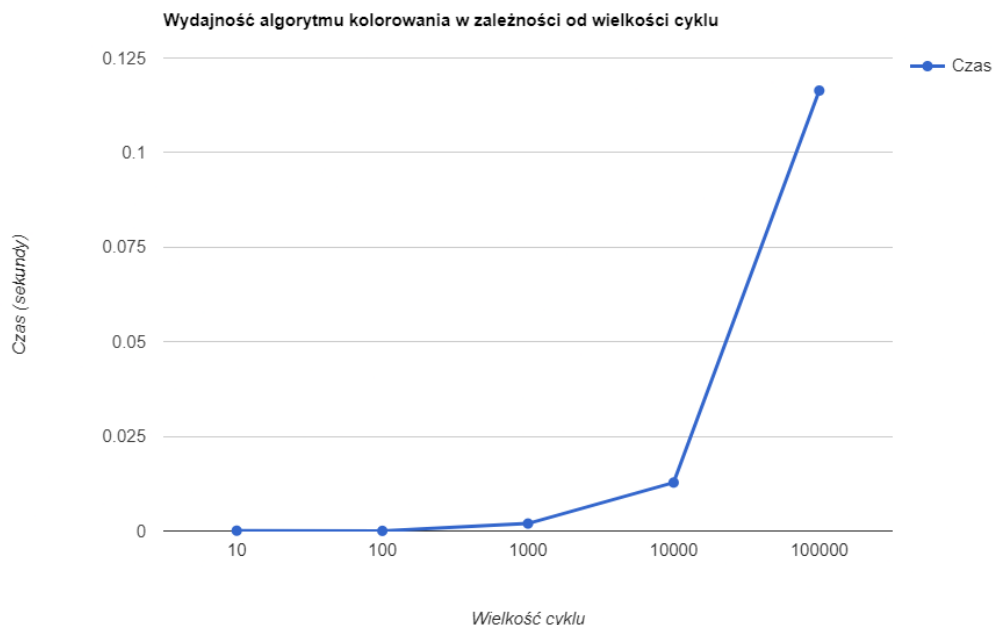
(d) Przykładowy komunikat o błędzie podczas obliczeń

5 Testy wydajności algorytmów

Każdy z testów wydajności został uruchomiony 100 razy, a następnie na podstawie otrzymanych wyników został obliczony średni czas wykonania algorytmu. Zostały utworzone odpowiednie wykresy na podstawie tych danych.

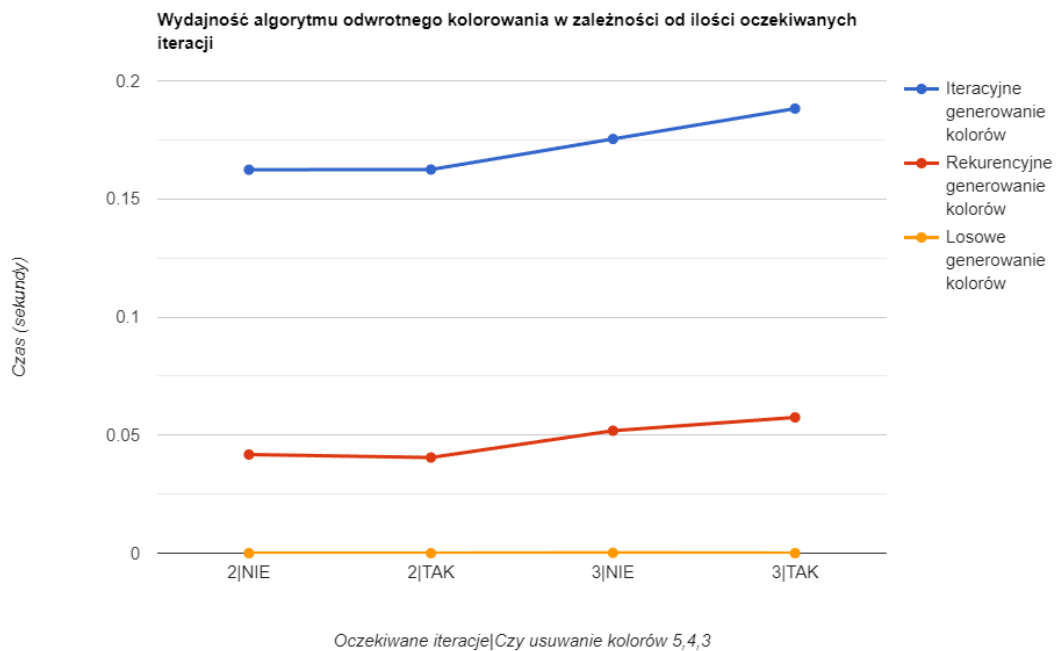
5.1 Test wydajności 3-kolorowania cyklu w zależności od wielkości cyklu

Początkowe kolory cykli zostały wybrane losowo z przedziału od 0 do 100.



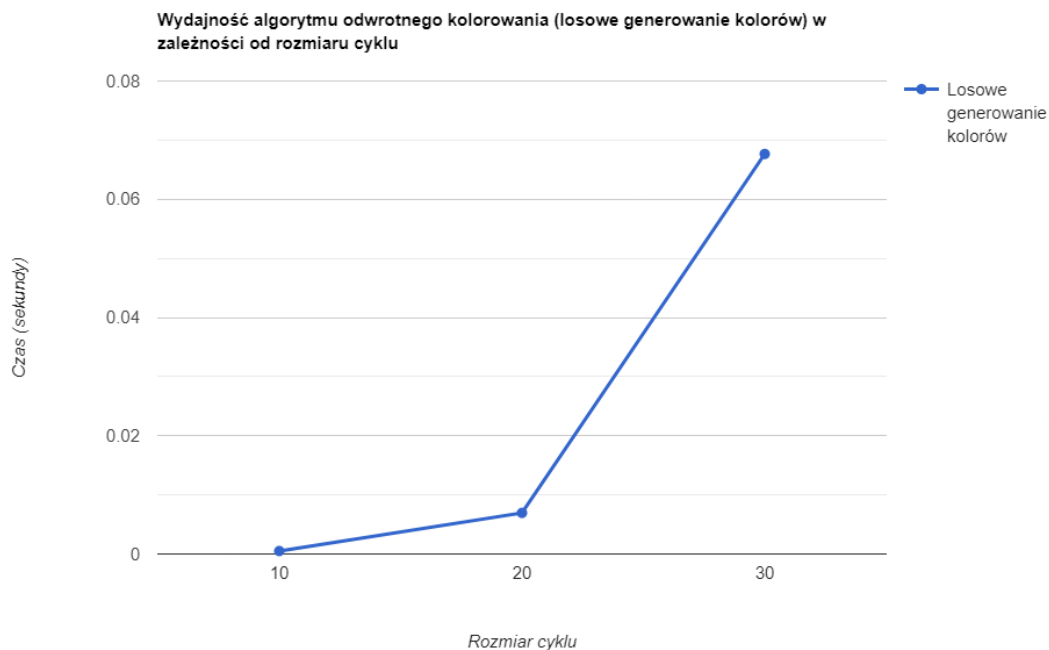
5.2 Test wydajności odwrotnego 3-kolorowania cyklu w zależności od oczekiwanej ilości iteracji

Przetestowane zostały wszystkie algorytmy odwrotnego 3-kolorowania cyklu. Testy zostały przeprowadzone dla cyklu wielkości 4. Potencjalne kolory początkowe wierzchołków zostały wybrane z przedziału od 0 do 30.



5.3 Test wydajności odwrotnego 3-kolorowania cyklu w zależności od wielkości cyklu

Przetestowany został algorytm z wykorzystaniem losowego generowania potencjalnych początkowych kolorów wierzchołków. Testy zostały przeprowadzone dla oczekiwanej liczby iteracji 3, bez usuwania liczb 5, 4, 3. Potencjalne kolory początkowe wierzchołków zostały wybrane z przedziału od 0 do 100.



6 Bibliografia

- Dziemiańczuk, M. (2024). Materiały z wykładu z przedmiotu Zaawansowane Algorytmy. Uniwersytet Gdański.
- <https://docs.oracle.com/javase/tutorial/uiswing/>
- <https://commons.apache.org/proper/commons-lang/>
- <https://www.rapidtables.com/>