Projekt 1

Matematyka dyskretna

Symulator gry worldie

Wykonał:

Ivan Kaliankovich

Spis treści

[Algorytm generowania podzbiorów zbioru 2](#_Toc153481803)

[Algorytm generowania k-elementowych podzbiorów zbioru 2](#_Toc153481804)

[Algorytm generowania permutacji zbioru 4](#_Toc153481805)

[Zadanie 4 4](#_Toc153481806)

# Algorytm generowania podzbiorów zbioru

Umieszczam poniżej kod fragment kodu zawierający algorytm  
Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie  
W pliku .m wołam algorytm i wypisuję jego zawartość w konsoli   
Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

W tym przypadku 7 to jest największy element zbioru głównego, podaję również pierwszy zbiór, od którego generowane są pozostaje zbiory.

# Algorytm generowania k-elementowych podzbiorów zbioru

Kod źródłowy algorytmu:

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Warunki początkowe:  
Obraz zawierający tekst, Czcionka, numer, biały

Opis wygenerowany automatycznie

# Algorytm generowania permutacji zbioru

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Warunki początkowe:  
Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

# Zadanie 4

Główną motywacją do zrobienia zadania był fakt, że czasem z kolegami lubimy grać w różnego rodzaju gry ze słowami. Wybrałem zgadywanie słów z przedstawionych liter jako najprostszy z przypadków.  
  
Za podstawę rozwiązywania zadania wybrałem nieco zmodyfikowany algorytm permutacji z zadania 3 powyżej.  
Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie  
Teraz funkcja permutację zwraca ostateczny zbiór po p iteracjach algorytmu oraz nic nie wypisuje.  
Gra zaczyna się od użytkownika, który jest proszony o wpisanie dowolnego słowa do konsoli. Po wpisaniu słowa wyskakuję okienko dla użytkownika 2 ze słowem o zamienionej kolejności liter w stosunku do słowa wprowadzonego przez użytkownika 1. Po zobaczeniu słowa, użytkownik 2 ma nieskończoną ilość prób do zgadnięcia oryginalnego słowa. W dowolnym momencie użytkownik może wpisać słowo kluczowe „pomoc”, wtedy pod permutowanym słowem pojawi nowe, pomocnicze, w którym pierwsze dwie litery są prawidłowe, natomiast reszta liter jest jeszcze raz permutowana.  
Obraz zawierający Czcionka, biały, Grafika, tekst

Opis wygenerowany automatycznie  
Tak wygląda interfejs w tym przypadku, początkowym słowem jest matematyka.  
Obraz zawierający Czcionka, tekst, biały, Grafika

Opis wygenerowany automatycznie  
W przypadku zgadnięcia słowa pojawia się komunikat „Zgadłeś”

Poniżej przedstawiam kod programu:  
Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie  
Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie  
Znaczącą wadą tego programu jest algorytm permutacji, który wymaga podania liczby iteracji permutacji oraz fakt, że algorytm jest bardzo zależny od macierzy wejściowej.   
W trakcie testów przekonałem się, że do takiego rodzaju zagadnienia bardziej pasującym jest rozwiązanie korzystające ze zwykłej funkcji randperm() tak jak poniżej  
Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie  
W tym przypadku słowo jest trudniejsze do zgadnięcia.  
Algorytm przedstawiony przeze mnie bardziej by się nadawał w przypadkach, kiedy wymagana jak największa ilość podzbiorów utworzonych ze zbioru początkowego. Na przykład program tworzący słowa złożonych z liter innego słowa wejściowego. Natomiast taki program wymagałby dużej bazy danych wszystkich słów.