Języki Skryptowe

dokumentacja projektu Randka olimpiada informatyczna XIX

Paweł Habrzyk, grupa 3E, Wydział Matematyki Stosowanej - Informatyka semeste 3

10 lutego 2021

Spis treści

1 Opis problemu przedstawionego w zadaniu

1.1 Treść zadania

Niech dana jest funkcja f(x) = yokreślona dla x ∈<a,b>i przyjmująca na tym przedziale dodatnie wartości.Chcemy znaleźć największą wartość tej funkcji w tym przedziale (zakładając, że nie znamy pojęcia pochodnej funkcji). Napisz program, który dla zadanej funkcji (można ograniczyć się do pewnego, znanego wcześniej zbioru funkcji) znajdzie jej maksimum.

1.2 Założenia

a)wartość taka może być osiągana na końcach przedziału (patrz rysunek, wykres zielony),

b)wartość taka może być osiągana wewnątrz przedziału (patrz rysunek, wykres czerwony),

c)wartość taka może być osiągana wielokrotnie w tym przedziale (patrz rysunek, wykres niebieski).

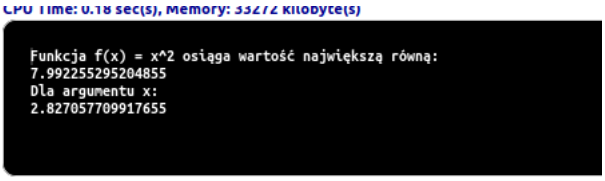
1.3 Przykład danych wejściowych i wyjściowych

input

a = 0
b = 3
f(x)=x
m = 50
n = 50

output

x = 2.8888, y = 2.888



2 Model Matematyczny

2.1 Analiza problemu

Sposób znalezienia tej wartości może być następujący:

1.Losujemy z przedziału <a,b> n liczb.

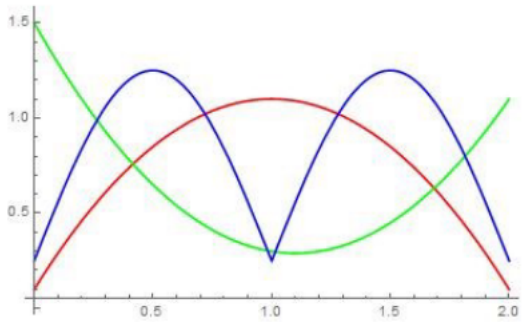
2.Dla każdej znich obliczamy wartość funkcji.

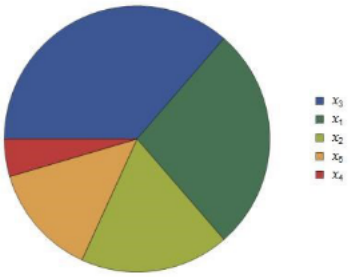
3.Ponieważ poszukujemy wartości największej –sortujemy malejąco wylosowane punkty wg. wartości funkcji.

4.Oczywiście nawet pierwszy z punktów nie musi gwarantować znalezienia wartości największej. Dlatego punkty te muszą „ewoluować": na podstawie listy punktów tworzyć będziemy kolejną listę, a ponieważ największe znaczenie mają te początkowe (po sortowaniu) punkty, dlatego one powinny „wydać na świat" (porównanie jest nieprzypadkowe –tak właśnie postępuje natura –osobniki najlepiej dostosowane wydają więcej potomstwa) statystycznie więcej „potomków". Odbywać się to będzie tak, że każdemu z punktów przypisywać będziemy odpowiednią wagę –proporcjonalną do wartości funkcji w tym punkcie (patrz „podział pizzy" na poniższym rysunku). Następnie, z przedziału <0,1> losujemy n liczb sprawdzając, do którego „kawałka pizzy" należy wylosowana liczba (proporcje podziału odcinka są takie same jak proporcje podziałupizzy). Po każdym takim losowaniu (a jest ich n) otrzymujemy osobnika takiego, jaki tworzył dany „kawałek pizzy". Populację sortujemy (malejąco po wartościach funkcji), następnie tak utworzona nowa populacja krzyżuje się(krzyżowanie punktów to ich średnia arytmetyczna): pierwszy z drugim, drugi z trzecim,..., przedostatni z ostatnim i, aby otrzymać w sumie n osobników, ostatni z pierwszym. Otrzymujemy w ten sposób pokolenie dzieci z pokolenia rodziców. Obliczamy wartości dla pokolenia dzieci i z obydwu pokoleń wybieramy 90 (z zaokrągleniem do najbliższej liczby naturalnej) z najlepszych osobników, a brakujące około 10 dolosowujemy z przedziału <a,b>.

5.Takie nowe pokolenie staje się teraz pokoleniem rodzicówi dla niego powtarzamy punkt 4 otrzymując kolejne pokolenie.

6.Czynności 4 –5 powtarzamy m razy i z ostatniego pokolenia wybieramy osobnika najlepszego, uznając go za rozwiązanie.





2.2 Instrukcja obsługi

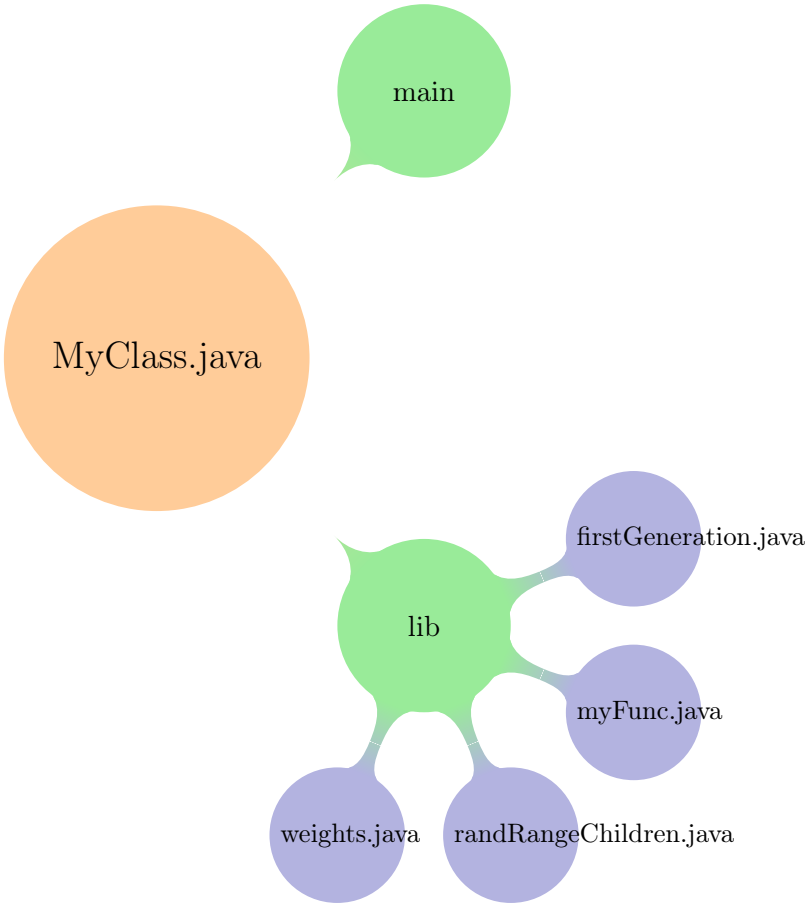
W celu uruchomienia programów projektowy ja osobiście przez brak dostępu do własnego komputera kod testowałem w dostępnym online kompilatorze. link: jdoodle.com/a/2YA2 (trzeba czasami kilka razy kliknąć execute)

2.3 Wymagania sprzętowe

System operacyjny Windows 10
Kompiloator java 1.8XXX

3 Implementacja

Opis, zasada i działanie programu ze względu na podział na pliki, nastepnie funkcje programu wraz ze szczegółowym opisem działania (np.: formie pseudokodu, czy odniesienia do równania)



MyClass.java zawiera głowny program wywołujacy pozostałe. firstgeneration.java wydaje na świat pierwsze pokolenie punktów a także losowe 10 procent punktów dodwane w każdej iteracji.Funkcja klasy randRangeChildern wydaje na świat "mieszanki"punktów wcześniejszych a qweights.java podaje wagi dla poszczególnych wyników na podtawie ich wartości (najwyższą wagę ma wartość największa). myFunc.java zawiera funkcję matematyczną (zachęcam do jej modyfikacji, pamiętając że w zadanych przedziałach, zgodnie z treścią zadania, funkcja musi być dodatnia)

4 Podsumowanie

4.1 Co zostało zrobione

Program pozwala obliczyć przybliżoną wartość maksiumum funkcji w podanym zakreaie.

4.2 Dalsze prace

Algorytm jest niewydajny, dobrym pomysłem wydaje się tego poprawa. Interfejs jest bardzo słaby, można to poprawić.