Problem 18: Zbiór Mandelbrota

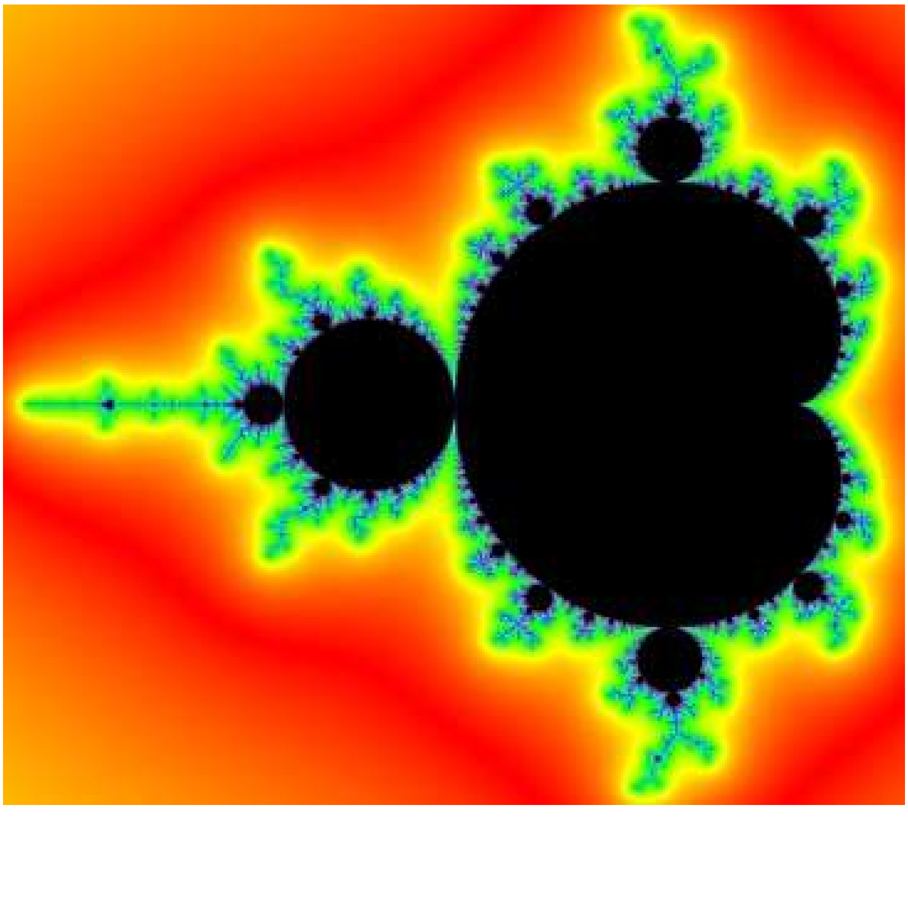
Punkty: 45

Autor: Louis Ronat, Denver, Kolorado, Stany Zjednoczone

Wprowadzenie do problemu

Zbiór Mandelbrota jest rysowany poprzez założenie funkcji rekurencyjnej $Z_{n+1} = {Z_n}^2 + c$, gdzie c to liczba zespolona formy $a + bi$ (w matematyce i to liczba urojona o wartości $\sqrt{-1}$; stąd $i^2 = -1$). Poprzez wielokrotną iterację, używając każdej wartości Z do obliczenia kolejnej wartości, dowiadujemy się, że w przypadku niektórych wartości początkowych c Z rośnie bez końca. W innych przypadkach Z pozostaje ograniczona.

Aby narysować zbiór Mandelbrota, korzystamy z „płaszczyzny zespolonej”, gdzie pozioma oś X reprezentuje wartość a, a pionowa oś Y reprezentuje wartość b. Każdy punkt ma swoją barwę zależną od liczby iteracji (n), jakie możemy wykonać, zanim wartość bezwzględna Z ($|Z_n|$) przekroczy określoną wartość. W takim momencie mówimy, że funkcja staje się rozbieżna (ulega dywergencji). Na poniższym obrazie barwa czarna oznacza, że $|Z_n|$ pozostała poniżej podanej wartości dla wszystkich wartości n. Niebieskie piksele reprezentują punkty, w których po wielu iteracjach $|Z_n|$ przekroczyła tę wartość; w przypadku czerwonych pikseli liczba iteracji była mniejsza.



Rozważmy funkcję korzystającą z wartości $c = 1.1 + 2i$.

Niezależnie od wartości c, wartość Z_0 pozostaje zawsze równa 0. Możemy z tego skorzystać w celu określenia wartości Z_1:

$$Z_1 = {Z_0}^2 + c$$
$$Z_1 = 0^2 + 1.1 + 2i$$
$$Z_1 = 1.1 + 2i$$

To pokazuje, że dla każdej wartości c jest $Z_1 = c$. Teraz musimy ustalić, czy funkcja stała się rozbieżna. Na potrzeby tego problemu uznajmy, że funkcja stała się rozbieżna, jeśli $|Z_n| \geq 100$. Ponieważ i to liczba urojona, użyjemy tego równania do ustalenia wartości bezwzględnej liczb formy $a + bi$:

$$|Z_1| = \sqrt{{a_1}^2 + {b_1}^2}$$
$$|Z_1| = \sqrt{1.1^2 + 2^2}$$
$$|Z_1| = \sqrt{1.21 + 4}$$
$$|Z_1| \approx 2.2825$$

2,2825 to mniej niż 100, zatem funkcja nie stała się jeszcze rozbieżna. Potrzebujemy więcej iteracji, by ustalić, czy w ogóle i kiedy funkcja stanie się rozbieżna:

$$Z_2 = {Z_1}^2 + c$$
$$Z_2 = (a_1 + b_1 i)^2 + a_0 + b_0 i$$
$$Z_2 = (1.1 + 2i)^2 + 1.1 + 2i$$
$$Z_2 = 1.1^2 + 1.1(2i) + 1.1(2i) + (2i)^2 + 1.1 + 2i$$
$$Z_2 = 1.21 + 4.4i - 4 + 1.1 + 2i$$
$$Z_2 = -1.69 + 6.4i$$
$$a_2 = -1.69$$
$$b_2 = 6.4$$
$$|Z_2| = \sqrt{-1.69^2 + 6.4^2}$$
$$|Z_2| \approx \sqrt{2.8561 + 40.96}$$
$$|Z_2| \approx 6.6194$$

(Należy pamiętać, że $i^2 = -1$, zatem powyżej $(2i)^2 = 2^2 * i^2 = 4 * -1 = -4$.)

$|Z_2|$ to ciągle mniej niż 100, zatem funkcja nie stała się jeszcze rozbieżna. Ilu iteracji potrzebujemy na dotarcie do tego miejsca?

n	***Z***	***a***	***b***	**|*Z*|**
1	1.1 + 2*i*	1.1	2	2.2825
2	-1.69 + 6.4*i*	-1.69	6.4	6.6194
3	-37.0039 - 19.632*i*	-37.0039	-19.632	41.8892
4	984.9732 + 1454.9211*i*	984.9732	1454.9211	1756.9769

Zatem przy $n = 4$ widzimy, że wartość $|Z| > 100$. Oznacza to, że dla tej wartości c funkcja stała się rozbieżna w 4 kroku iteracji. Barwimy punkt o współrzędnych x = 1,1 i y = 2 kolorem odpowiednim dla tej wartości i przechodzimy do następnej sprawdzanej wartości.

Opis problemu

Wasz program musi zidentyfikować barwę używaną do renderowania zbioru Mandelbrota dla zadanej wartości c. Z pomocą poniższej tabeli i wyjaśnień podanych powyżej należy ustalić, jakiej barwy użyć:

Wartość n, gdy funkcja staje się rozbieżna	Barwa
≤ 10	RED (CZERWONA)
11-20	ORANGE (POMARAŃCZOWA)
21-30	YELLOW (ŻÓŁTA)
31-40	GREEN (ZIELONA)
41-50	BLUE (NIEBIESKA)
≥ 51	BLACK (CZARNA)

W przykładowych obliczeniach funkcja staje się rozbieżna w $n = 4$, zatem prawidłowa barwa dla tej wartości c to czerwona.

Przykładowe dane wejściowe

Pierwszy wiersz danych wejściowych waszego programu, **otrzymanego przez standardowe wejście**, będzie zawierać dodatnią liczbę całkowitą oznaczającą liczbę przypadków testowych. Każdy przypadek testowy będzie zawierać pojedynczy wiersz składający się z dwóch liczb dziesiętnych oddzielonych spacjami. Te liczby to odpowiednio wartości a i b. Należy pamiętać, że $c = a + bi$.

```
4
1.1 2.0
-0.7 0.2
-0.5 0.65
-0.5 0.608
```

Przykładowe dane wyjściowe

W każdym przypadku testowym wasz program powinien wyświetlić wartość c, po której następuje spacja, a następnie barwa użyta do renderowania tej wartości, zgodnie z powyższą tabelą. Barwę należy wyświetlić wielkimi literami. Wartości dziesiętne powinny być wyświetlone tak, jak zostały podane w danych wejściowych.

```
1.1+2.0i RED
-0.7+0.2i BLACK
-0.5+0.65i ORANGE
-0.5+0.608i BLUE
```