

Sprawozdanie 1

Mandelbrot

Andrzej Żwirko
55575

1. Wstęp

Sprawozdanie dotyczy analizy równoległego generowania zbioru Mandelbrota, testując trzy różne strategie podziału pracy: blokowy, roleta oraz dynamiczny z mutexem. Porównanie obejmuje dwa procesory: Apple Silicon M2 Pro i Intel i7-8700K.

2. Specyfikacja techniczna procesorów

1. Apple Silicon M2 Pro: wielordzeniowy ARM, wysoka przepustowość RAM, duża liczba wątków sprzętowych.

- 10-core (6 performance, 4 efficiency)

2. Intel i7-8700K: 6 rdzeni, obsługa HyperThreadingu, wysoka częstotliwość zegara, klasyczna architektura x86.

- Total Cores 6

- Total Threads 12

- Max Turbo Frequency 4.70 GHz

- Processor Base Frequency 3.70 GHz

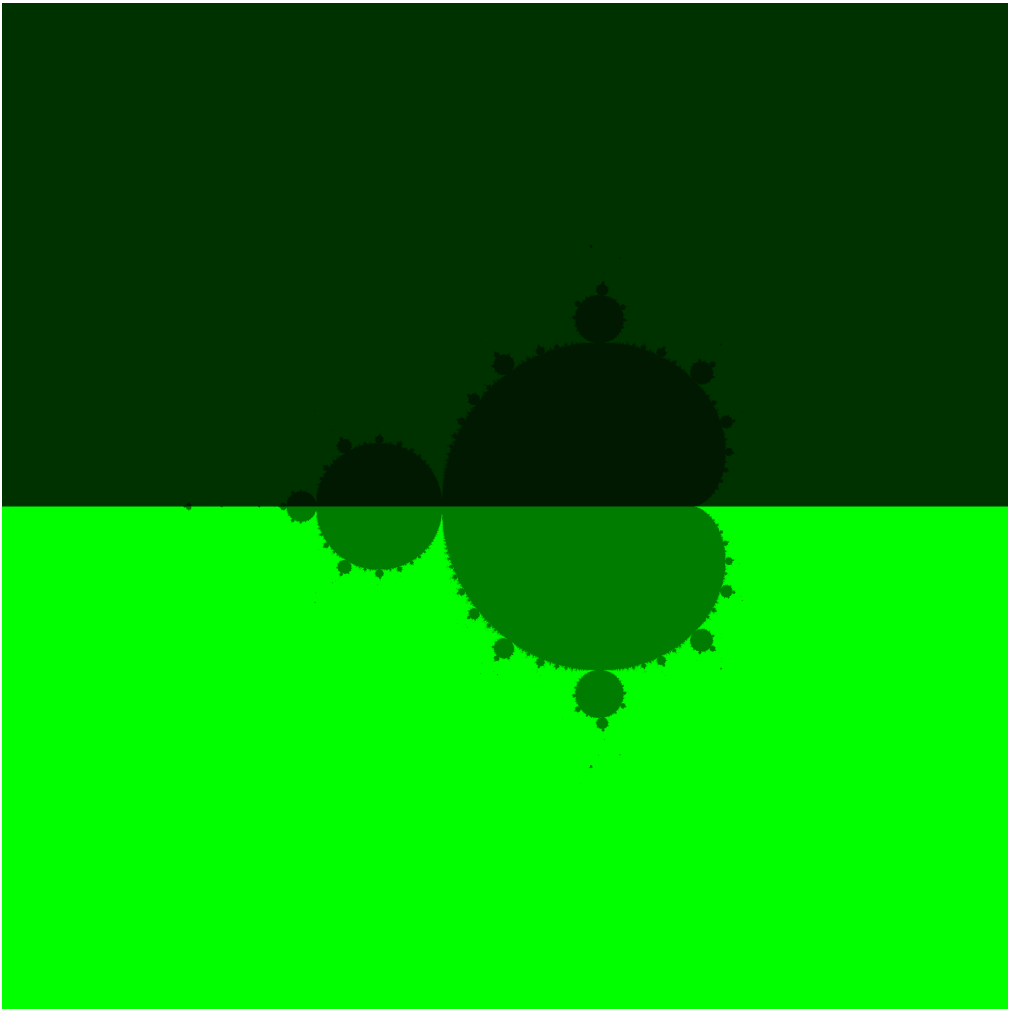
- Cache 12 MB Intel® Smart Cache

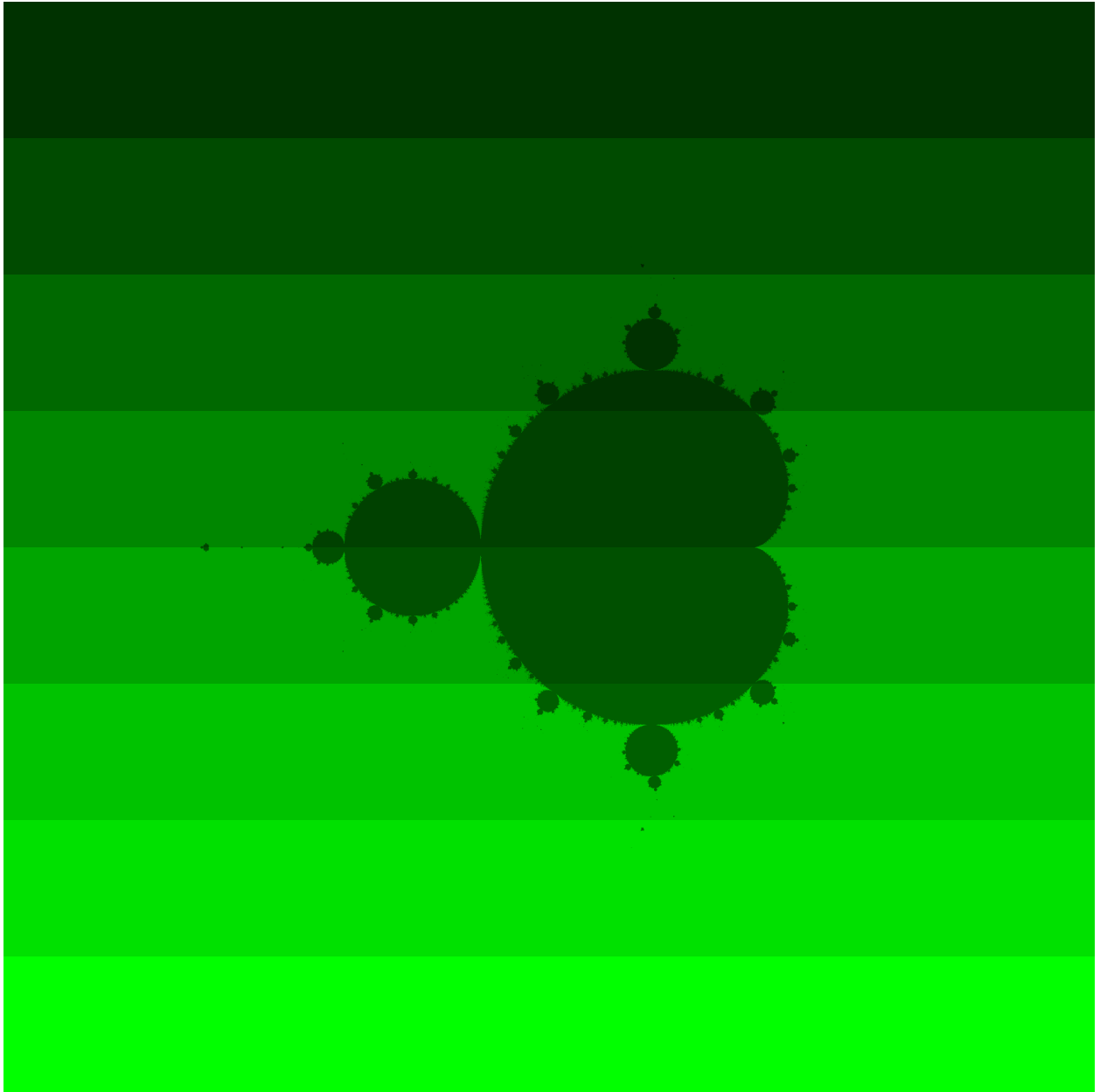
3. Wyniki czasowe i zdjęcia wynikowe

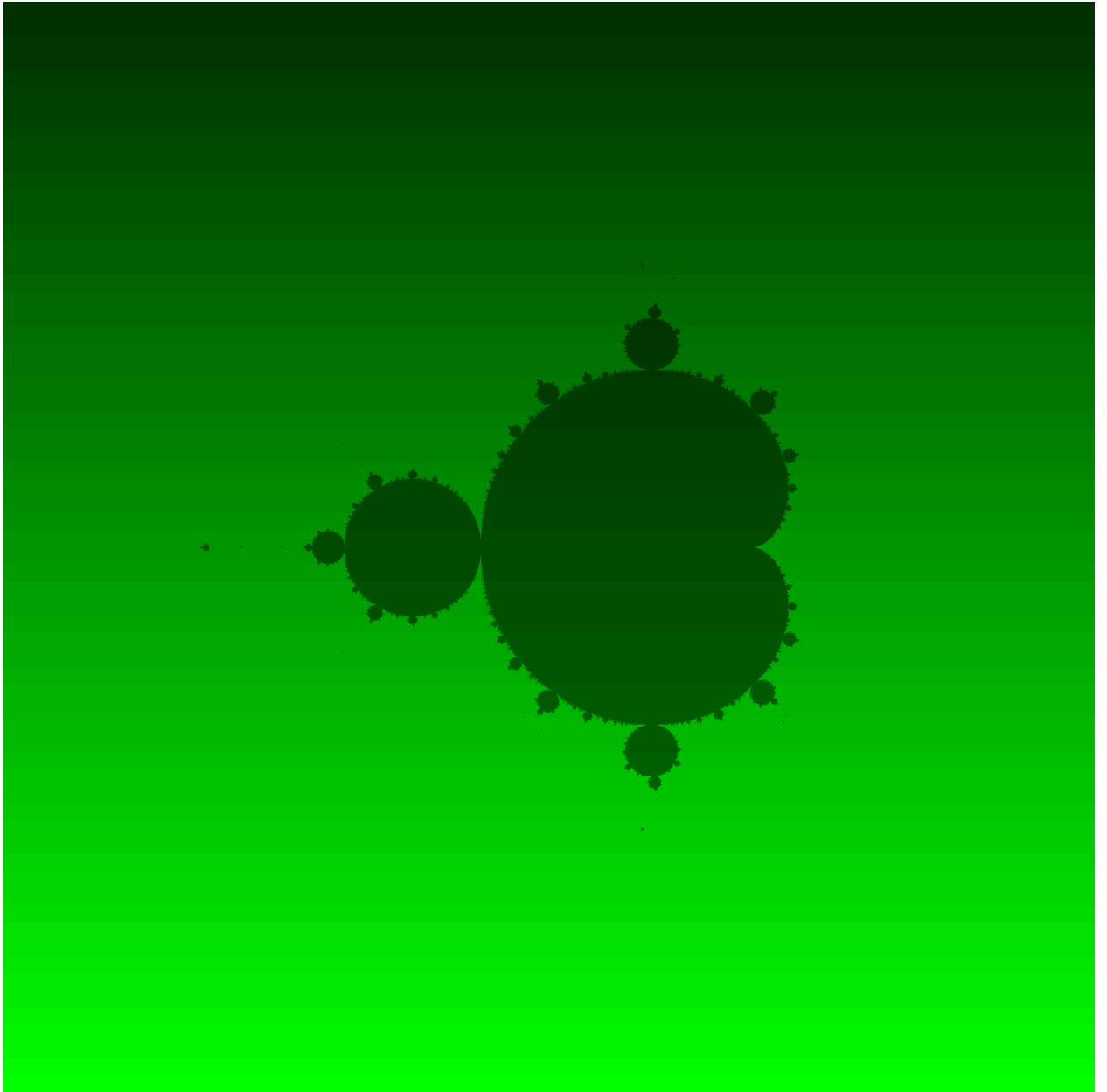
1. Blokowo

Apple Silicon M2 Pro			
Wątki	Czas	Max wątków	Min wątków
2	1.855s	1: 451702546 (50.04%)	0: 450979589 (49.96%)
4	1.163s	2: 433443582 (48.02%)	0: 18246087 (2.02%)
8	0.949s	4: 344716569 (38.19%)	0: 7438787 (0.82%)
16	0.585s	8: 205657116 (22.78%)	0: 3395567 (0.38%)
32	0.383s	16: 107743708 (11.94%)	0: 1577699 (0.17%)
64	0.334s	32: 55322634 (6.13%)	0: 745329 (0.08%)
128	0.316s	64: 27982067 (3.10%)	0: 357027 (0.04%)

Intel i7-8700k			
Wątki	Czas	Max wątków	Min wątków
2	2.625s	1: 451702703 (50.04%)	0: 450979746 (49.96%)
4	1.356s	2: 433443585 (48.02%)	0: 18246087 (2.02%)
8	1.066s	4: 344716569 (38.19%)	7: 7443020 (0.82%)
16	0.653s	8: 205657116 (22.78%)	0: 3395567 (0.38%)
32	0.399s	16: 107743708 (11.94%)	0: 1577699 (0.17%)
64	0.325s	32: 55322634 (6.13%)	0: 745329 (0.08%)
128	0.324s	64: 27982067 (3.10%)	0: 357027 (0.04%)



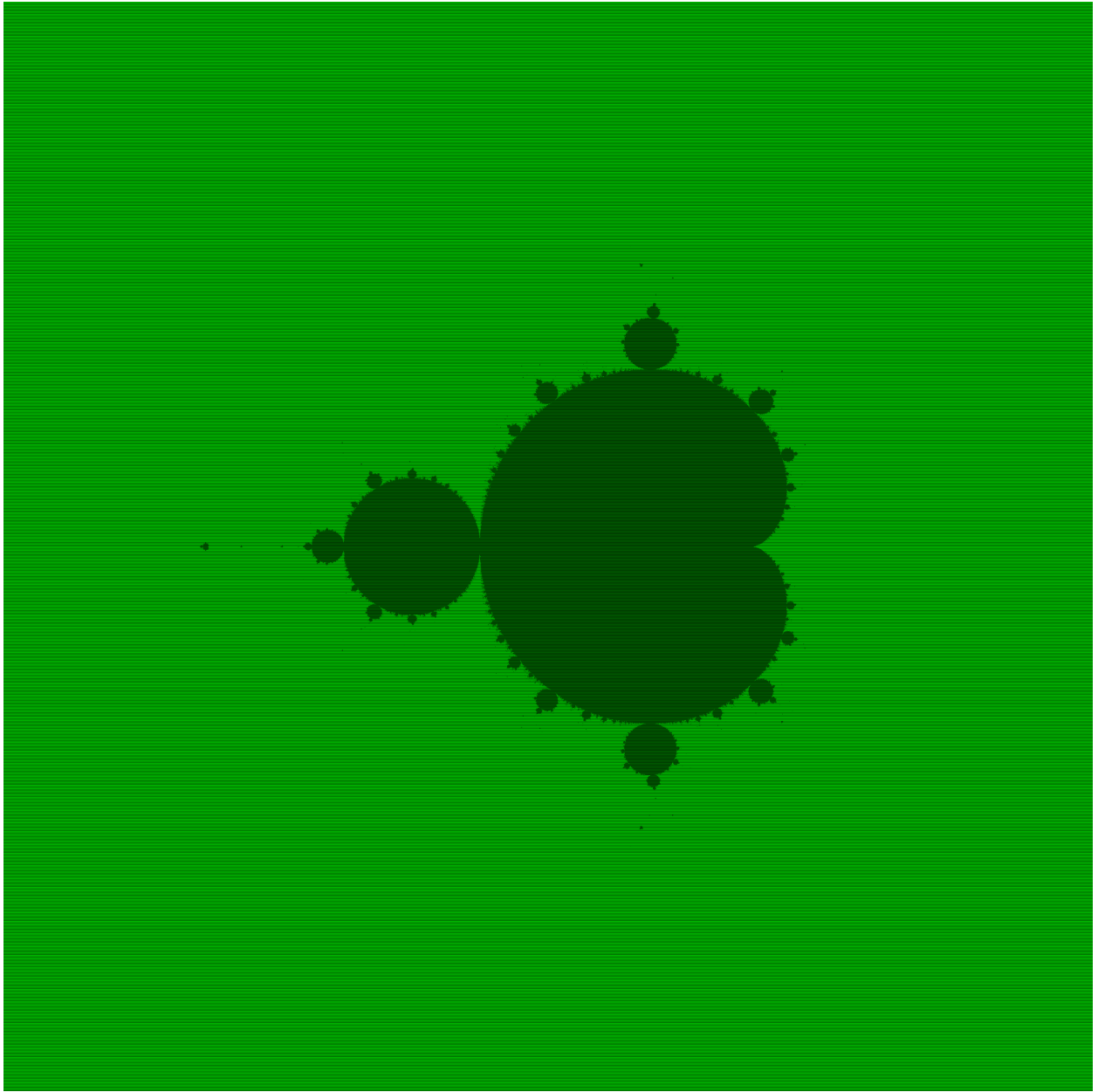


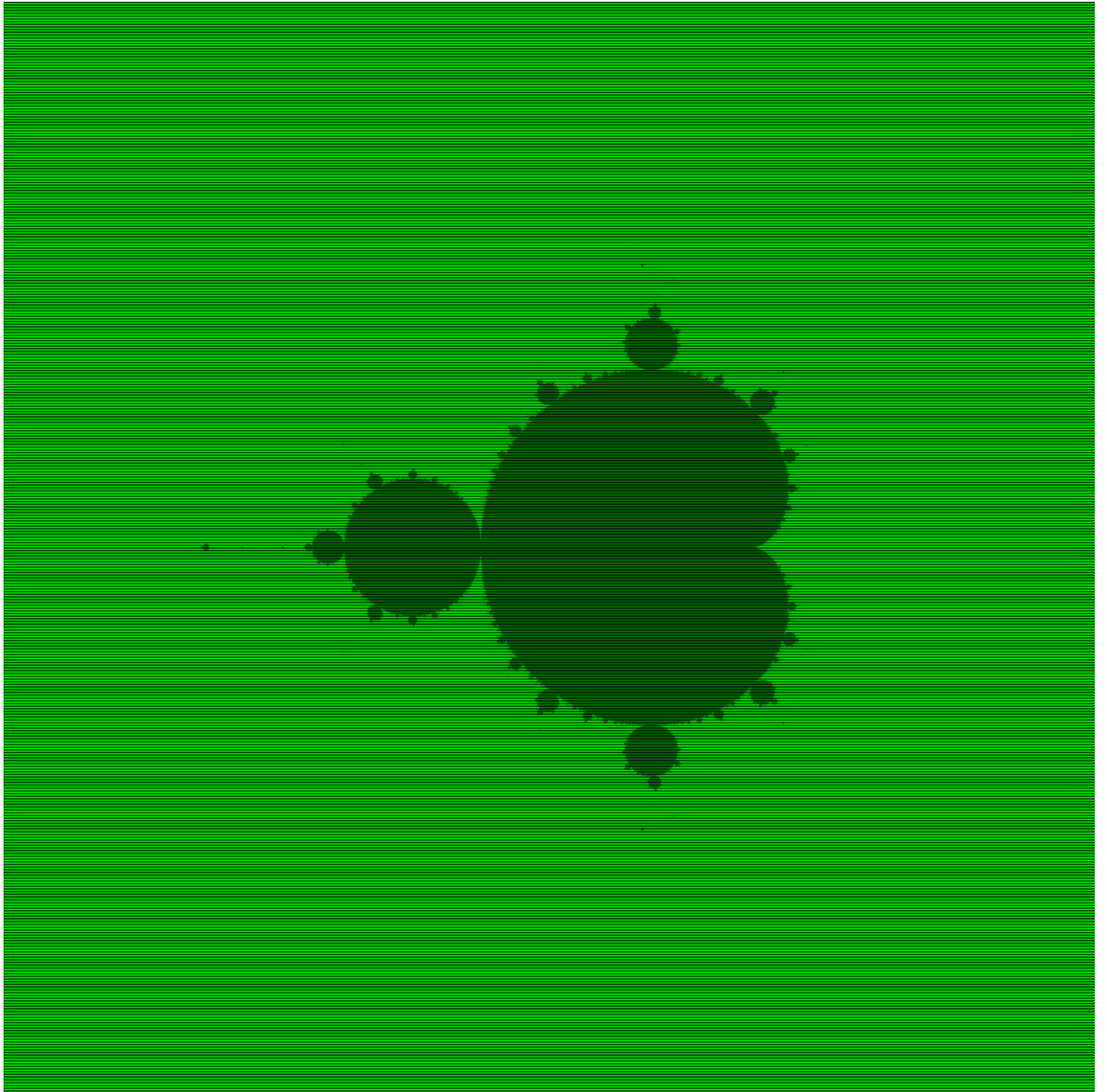


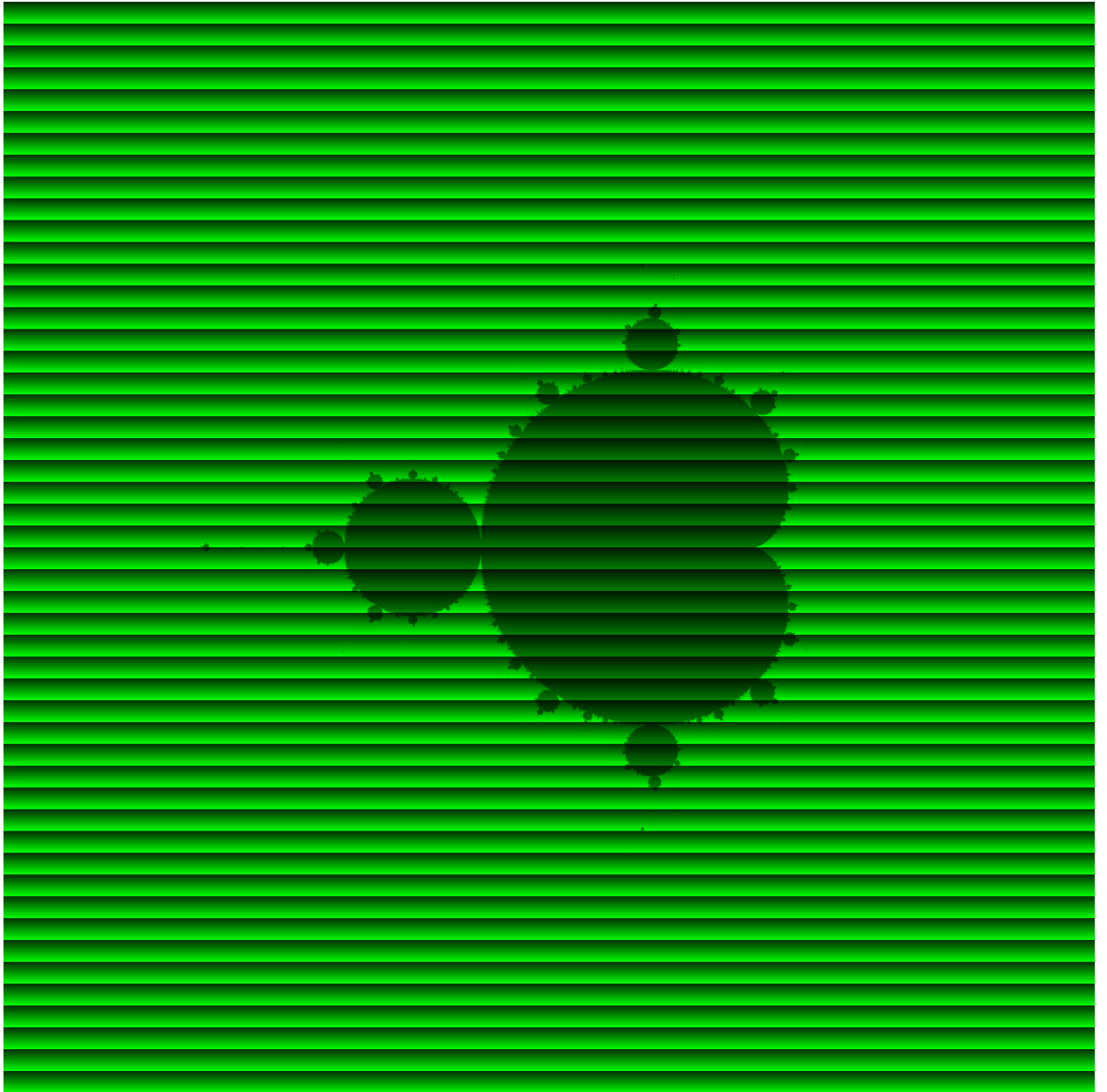
2. Roleta

Apple Silicon M2 Pro			
Wątki	Czas	Max wątków	Min wątków
2	2.336s	0: 451417355 (50.01%)	1: 451264780 (49.99%)
4	0.619s	0: 225789138 (25.01%)	2: 225628217 (25.00%)
8	0.353s	0: 112996350 (12.52%)	4: 112792788 (12.50%)
16	0.297s	0: 56569552 (6.27%)	13: 56388545 (6.25%)
32	0.301s	0: 28333548 (3.14%)	3: 28184620 (3.12%)
64	0.299s	0: 14261168 (1.58%)	32: 14072380 (1.56%)
128	0.299s	0: 7261400 (0.80%)	63: 6996717 (0.78%)
512	0.295s	128: 1848960 (0.20%)	99: 1656234 (0.18%)

Intel i7-8700k			
Wątki	Czas	Max wątków	Min wątków
2	2.667s	0: 451417667 (50.01%)	1: 451264782 (49.99%)
4	0.697s	0: 225789446 (25.01%)	2: 225628221 (25.00%)
8	0.375s	0: 112996658 (12.52%)	4: 112792788 (12.50%)
16	0.322s	0: 56569860 (6.27%)	3: 56388547 (6.25%)
32	0.305s	0: 28333856 (3.14%)	3: 28184620 (3.12%)
64	0.290s	0: 14261476 (1.58%)	32: 14072380 (1.56%)
128	0.275s	0: 7261400 (0.80%)	63: 6996717 (0.78%)
512	0.296s	128: 1848960 (0.20%)	99: 1656234 (0.18%)



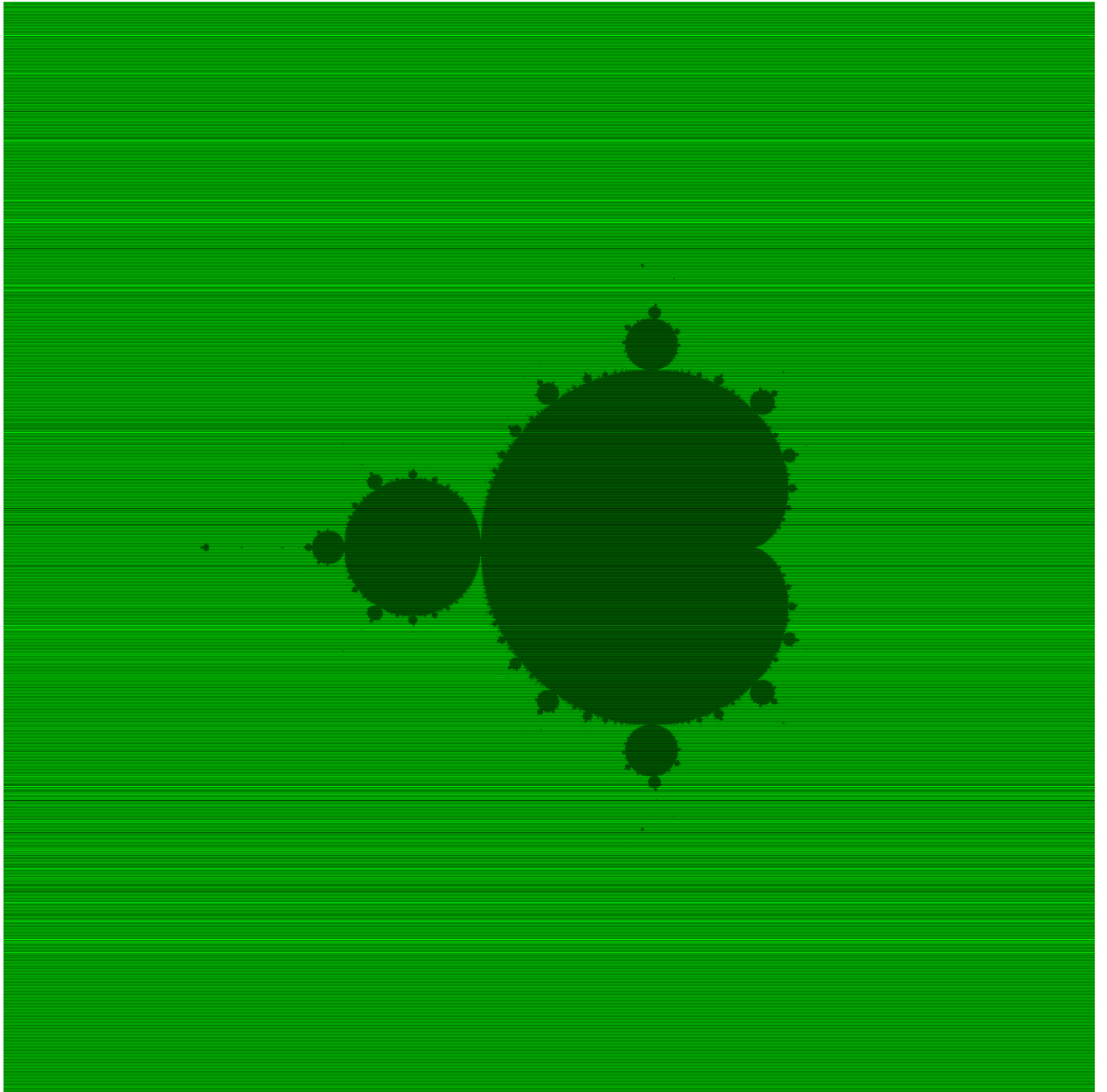


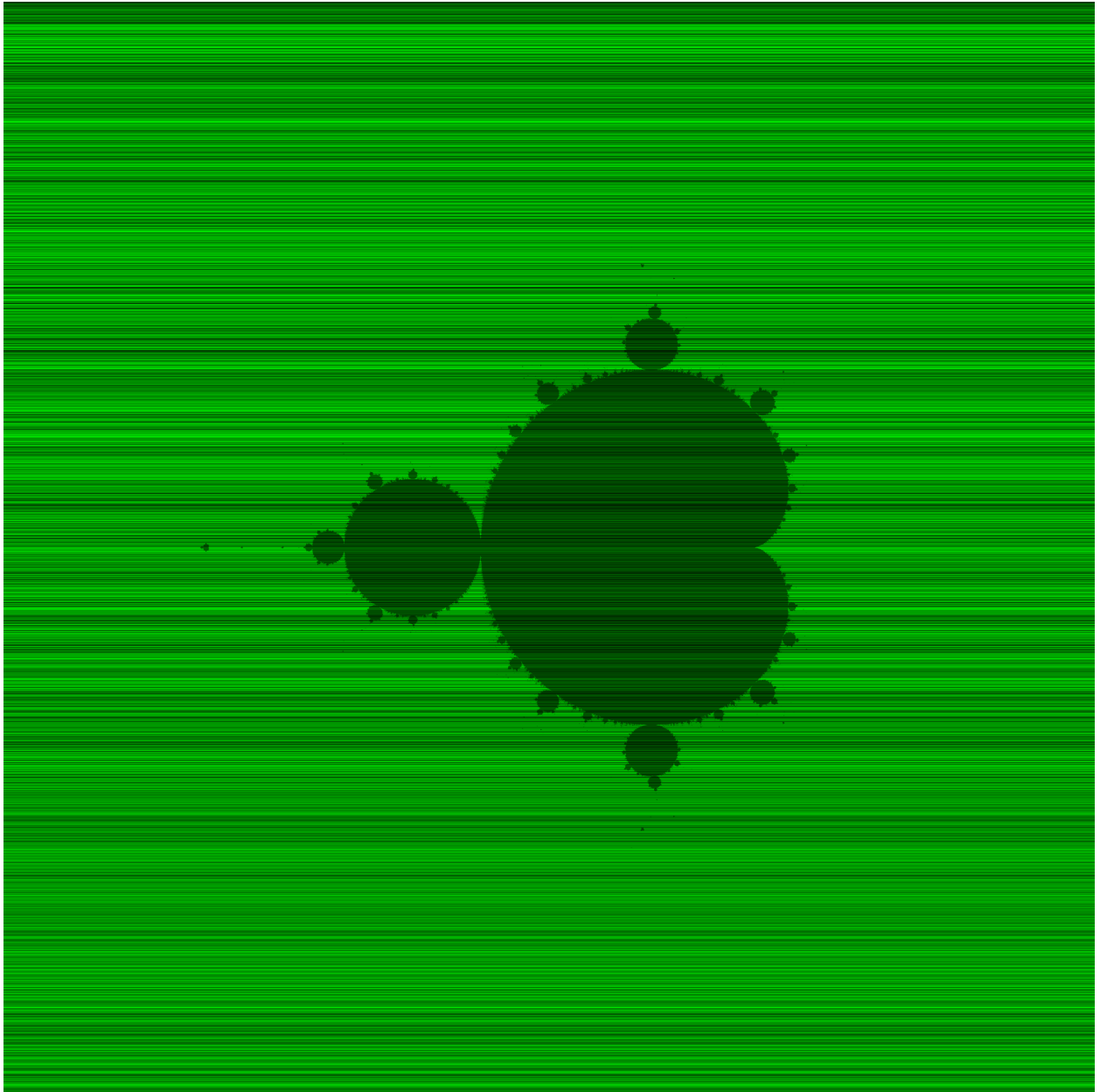


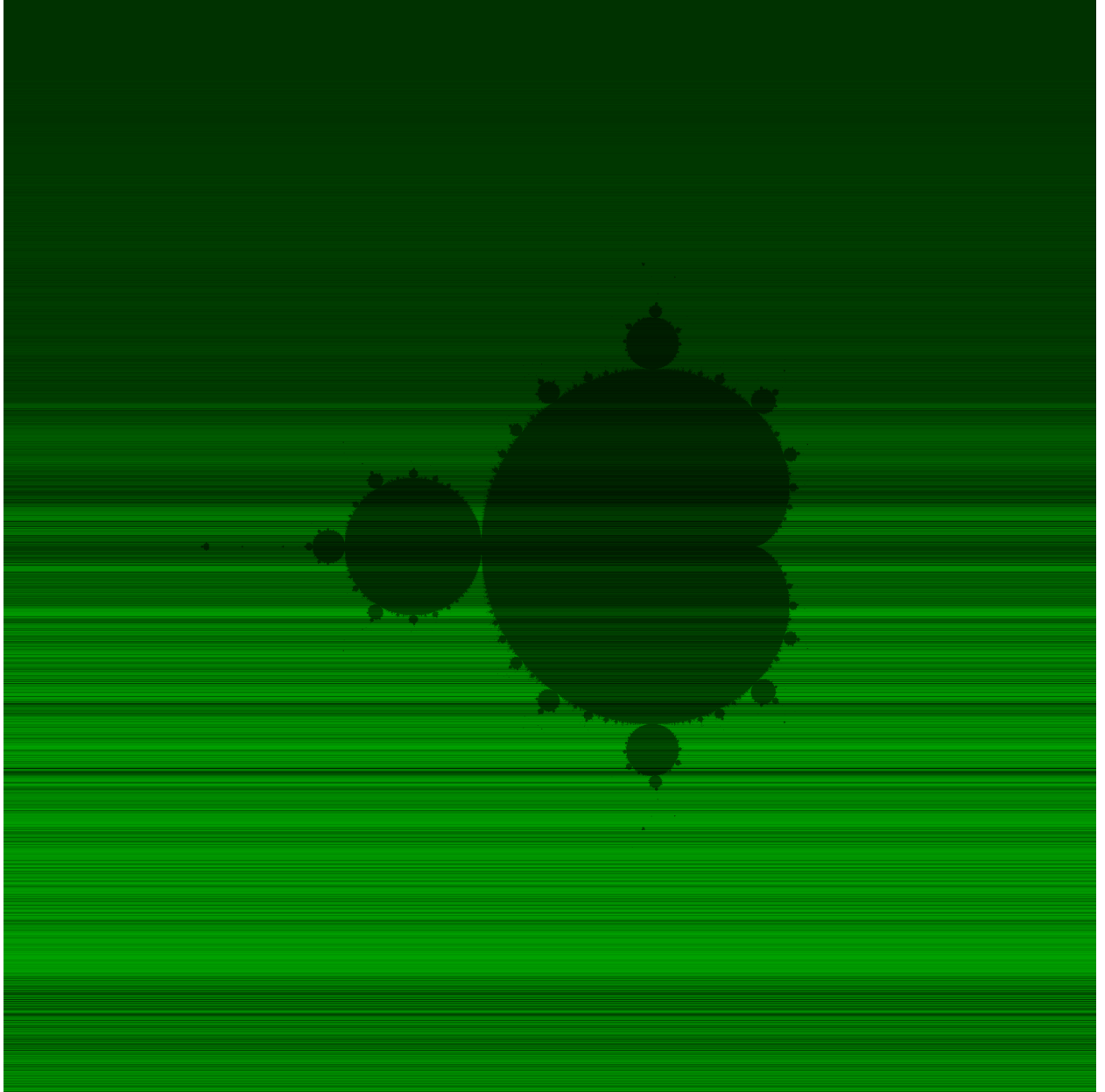
3. Mutex

Apple Silicon M2 Pro			
Wątki	Czas	Max wątków	Min wątków
2	2.322s	1: 451772776 (50.05%)	0: 450909359 (49.95%)
4	0.604s	2: 226313701 (25.07%)	0: 224829086 (24.91%)
8	0.340s	3: 116306630 (12.88%)	6: 107970120 (11.96%)
16	0.296s	8: 86125261 (9.54%)	11: 50558926 (5.60%)
32	0.296s	27: 31883174 (3.53%)	4: 24111646 (2.67%)
64	0.301s	23: 19950127 (2.21%)	37: 10885382 (1.21%)
128	0.302s	105: 11581413 (1.28%)	122: 3701986 (0.41%)
512	0.298s	18: 11533344 (1.28%)	265: 0 (0.00%)

Intel i7-8700k			
Wątki	Czas	Max wątków	Min wątków
2	2.618s	0: 452245791 (50.10%)	1: 450436658 (49.90%)
4	0.678s	3: 226242328 (25.06%)	2: 224809585 (24.90%)
8	0.374s	6: 116993136 (12.96%)	7: 108457099 (12.01%)
16	0.266s	6: 78122664 (8.65%)	15: 33754221 (3.74%)
32	0.263s	6: 70646098 (7.83%)	15: 0 (0.00%)
64	0.267s	7: 31414652 (3.48%)	33: 0 (0.00%)
128	0.285s	4: 76055915 (8.43%)	16: 0 (0.00%)
512	0.284s	4: 77682054 (8.61%)	12: 0 (0.00%)







4. Porównanie Architektur Procesorów

- Na M2 Pro czasy redukują się szybciej przy dużej liczbie wątków, co wynika z większej liczby rdzeni logicznych i lepszych algorytmów zarządzania wątkami w ARM.
- Intel i7-8700K nieco gorzej skaluje się powyżej liczby rdzeni, ale również korzysta z HyperThreadingu, co pozwala na lepsze wykorzystanie podziału roletowego oraz dynamicznego.

5. Kluczowe Wnioski

1. Dlaczego czasy redukują się pomimo liczby wątków > liczba CPU?

Rozkład pracy na więcej wątków niż sprzętowych rdzeni umożliwia lepsze wykorzystanie zasobów procesora przez przełączanie kontekstu i ukrywanie kosztów opóźnień, zwłaszcza przy zadaniach o nierównomierniej złożoności per-piksel/fractal. Prawo Amdahla pokazuje, że skalowanie przyspieszenia jest ograniczone przez część wykonywaną sekwencyjnie (inicjalizacja, zapis pliku, synchronizacja), ale im większa liczba wątków, tym lepiej można zrównoleglić część rzeczywistą obliczeń. Zwiększenie wątków zmniejsza czas aż do momentu zatorów i narzutu na zarządzanie.

2. Podział "roleta"

Podział roletowy (ang. stripe/round robin) oznacza przypisanie iteracji według wzoru: `for(iY=tid; iY<iYmax; iY+=nrt)`, gdzie każdy wątek bierze co n-ty krok. Pozwala to równomiernie rozłożyć pracę, nawet jeśli poszczególne obliczenia są różnej trudności — w przeciwieństwie do podziału blokowego każdy wątek rozkłada obliczenia na całą tablicę, niwelując skrajne obciążenia.

W praktyce zapewnia lepszą równowagę, szczególnie przy generowaniu fraktali o nierównomiernym rozkładzie iteracji.

3. Podział dynamiczny, mutex i sekcja krytyczna

W algorytmie dynamicznym każdy wątek bierze "wolny" blok danych przez mechanizm typu work-pool, co reguluje dostęp do zasobów przez mutex (mutual exclusion). Sekcja krytyczna to fragment kodu, do którego dostęp w danym momencie może mieć tylko jeden wątek — chroniony przez mutex. Zmienna sum iteracji obrazuje liczbę przetworzonych bloków — jej zmienność wynika z dynamicznego przydziału pracy i kolizji w pobieraniu z globalnej puli, stąd za każdym uruchomieniem są inne wartości (algorytm równoważy obciążenie na bieżąco, stąd zmienność sum w tablicy wyników).

Wyjaśnienie mutex: to obiekt synchronizacyjny, który umożliwia blokowanie sekcji kodu dla pojedynczego wątku.

6. Rekomendacje

- Optymalizować stosowany podział zależnie od architektury procesora — przy ARM zaleca się roletowy lub dynamiczny, przy x86 dynamiczny dla dużych obrazów.
- Unikać nadmiernej ilości synchronizacji, szczególnie przy wysokiej liczbie wątków.
- Większa liczba wątków efektywnie wykorzystuje zasoby, ale przy zbyt dużej liczbie narzut synchronizacji (mutex/sekcja krytyczna) może hamować przyspieszenie.
- Podział roletowy oferuje dobrą równowagę obciążenia w stosunku do podziału blokowego.
- Podział dynamiczny minimalizuje nierówne rozłożenie pracy, ale wprowadza dodatkowy narzut synchronizacji.