Timothée Guerin
260447866

Comp 251 Assignment 1

Description of algorithm

The $findCircles$ algorithm is looping the x and y coordinates from $minr$ to $n - minr$ then for all radius between $minr$ and $maxr$. Then it's calling $findCircle$ with the current x, y position and radius.

The $findCircle$ function first checking for the cardinal point to be of the good color, then for the cardinal point at $r + 1$ and $r - 1$(This accelerate the process of finding potential intersections, the intersections functions won't be called, especially for all the white circle that can be detected when the background is white). Then the function is looping using the Bresenham's algorithm. If the $findCircle$ function return $true$ then we call the intersect function for $r + 1$ and $r - 1$ to check for any touching circle.
If the intersect both return false then we can consider there is a circle and so we can add it into the array (ArrayList of array of size 3 (x, y, r))

Question 2

Question 2.a

As the Bresenham's algorithm takes time $2\pi r$. The findCircle and intersect functions function also take $2\pi r$. Then 0as we are calling findCircle once and intersect twice we have $6\pi r$. Then we have $6\pi r * (n - 2r)(n - 2r) = 6\pi r * (n^2 - 4nr + 4r^2)$

$$6\pi r * (n^2 - 4nr + 4r^2) \leq 6\pi r * (n^2 + 4r^2) = 6\pi r n^2 + 24\pi r\text{^}3$$

This gives us $O(n^2)$

Question 2.b

The best case is when the $findCircle$ return false just by checking the cardinal point. Then the $findCircle$ is $O(1)$. Then as were looping twice the time depends on $(n - 2r)^2$ which is $O(n^2)$.

Question 2.c

The time of the algorithm is

$$f(n, r) = 6\pi r * (n - 2r)^2$$

Let solve the derivative for r

$$\frac{d(f(n,r))}{d(r)} = 2\pi\ (n^2 - 8nr + 12r^2)$$

Solving to 0 gives us either

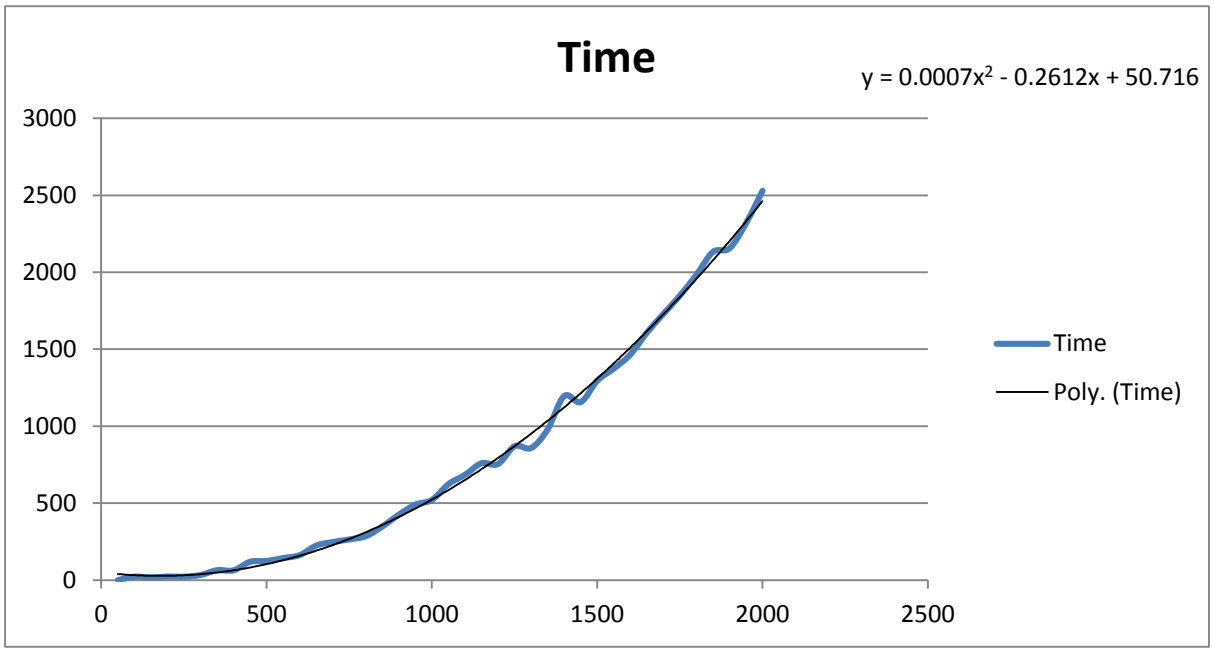
$$r = \frac{n}{2}\ or\ r = \frac{n}{6}$$

- $\frac{n}{2}$ Minimum is reach, for all $r \geq \frac{1}{2}$ the time is 0
- $\frac{n}{6}$ Is the maximum.

Then the maximum is reach for $r = \frac{n}{6}$

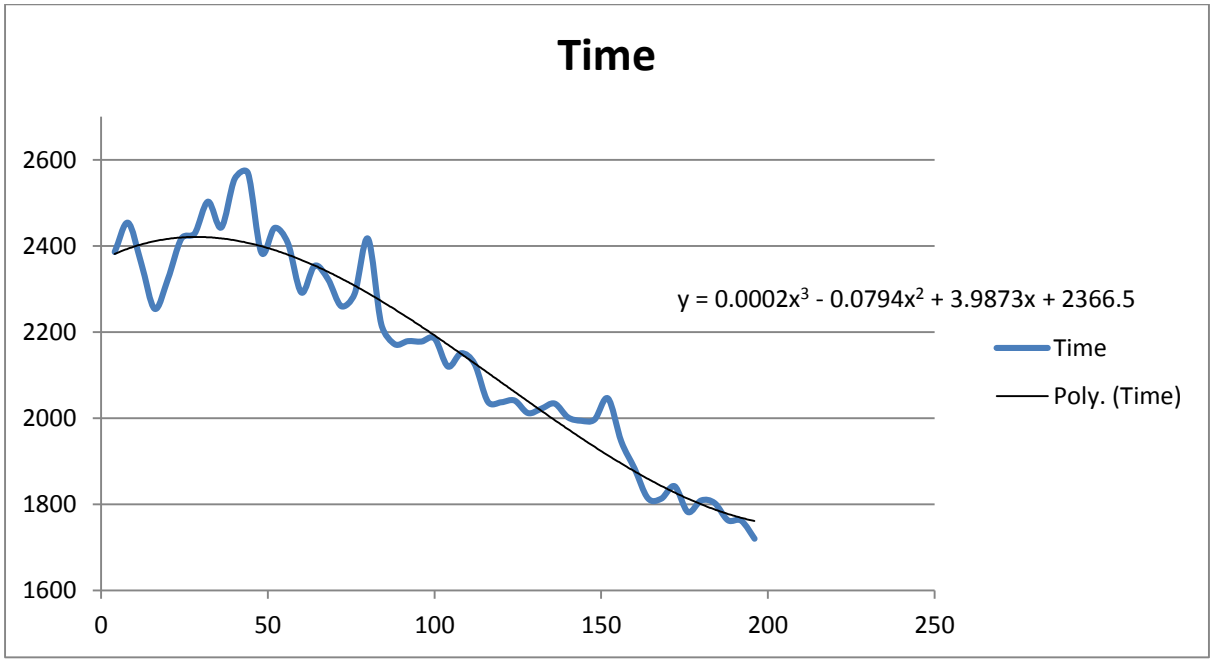
Question 3

Question 3.a



As we can see the time is depending on n^2 which again validate the previously found complexity of $O(n^2)$

Question 3.b



As we can see the function depends on r^3 which validate the previously found complexity if it depends on r:

$$6\pi r * (n - 2r)^2 * r = O(r^3)$$

Question 4

The current algorithm is only detecting circle traced using the Bresenham's algorithm. We can add an ϵ around the expected position of the pixel. We can check for all pixel touching the one we are currently checking if any of those have to good colors then we can consider the test to work. We also need to remove the inner intersection so it includes discs.

Alternatively we can use a loop with a θ going from 0 to 2π and by taking the cosines and sinus for x and y then by checking first for a circle then for possible ellipsis with $r + 1$ or $r - 1$ of radius.

Annexe

Graph 1

Radius	Time	Radius	Time
4	2386	104	2120
8	2454	108	2151
12	2359	112	2126
16	2254	116	2038
20	2324	120	2037
24	2417	124	2041
28	2429	128	2012
32	2503	132	2022
36	2443	136	2034
40	2556	140	2002
44	2569	144	1994
48	2385	148	1997
52	2442	152	2046
56	2406	156	1946
60	2292	160	1883
64	2354	164	1814
68	2323	168	1813
72	2260	172	1842
76	2289	176	1782
80	2417	180	1809
84	2218	184	1803
88	2172	188	1763

Graph 2

Size	Time	Size	Time
50	0	1050	623
100	22	1100	684
150	17	1150	759
200	23	1200	754
250	22	1250	870
300	33	1300	858
350	64	1350	979
400	63	1400	1196
450	119	1450	1156
500	124	1500	1295
550	143	1550	1374
600	162	1600	1465
650	224	1650	1607
700	247	1700	1729
750	264	1750	1848
800	285	1800	1983
850	348	1850	2132
900	425	1900	2155
950	490	1950	2318
1000	521	2000	2529

92	2179	192	1762	
96	2178	196	1720	
100	2185			