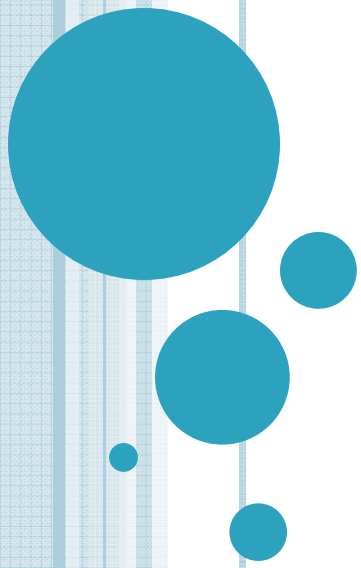


CS3241 COMPUTER GRAPHICS

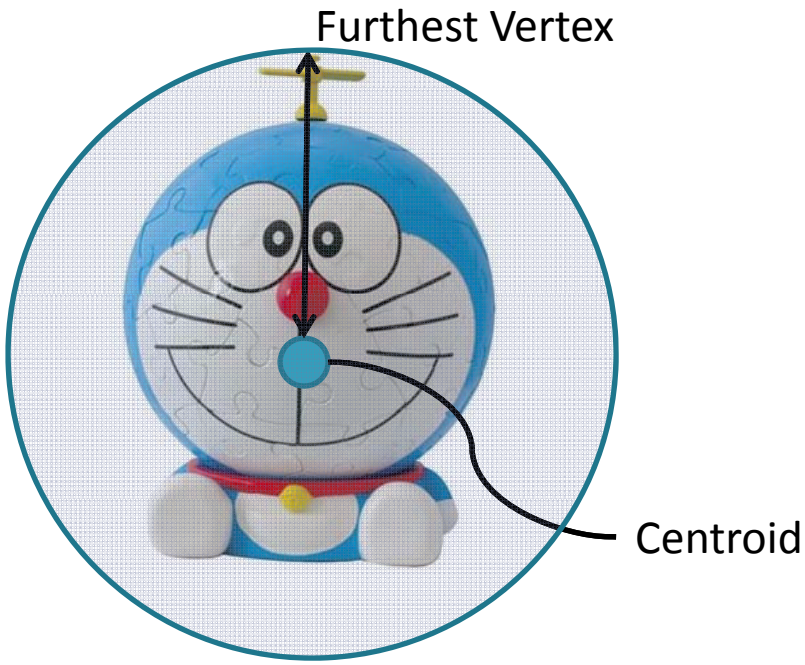
Tutorial 8

QUESTION #1

- Given an object represented in polygons, how to find its bounding sphere?

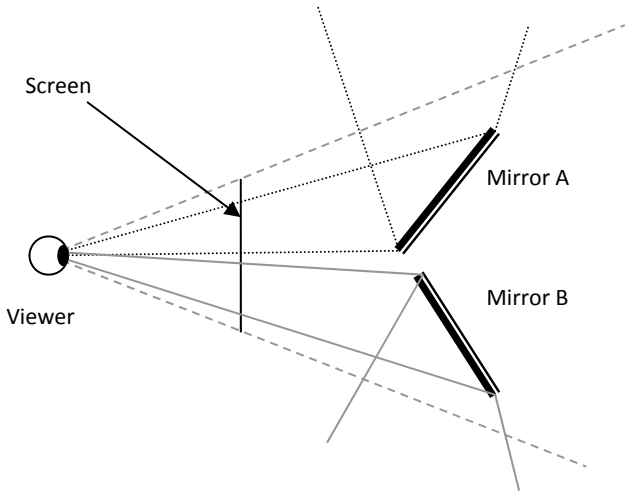
QUESTION #1

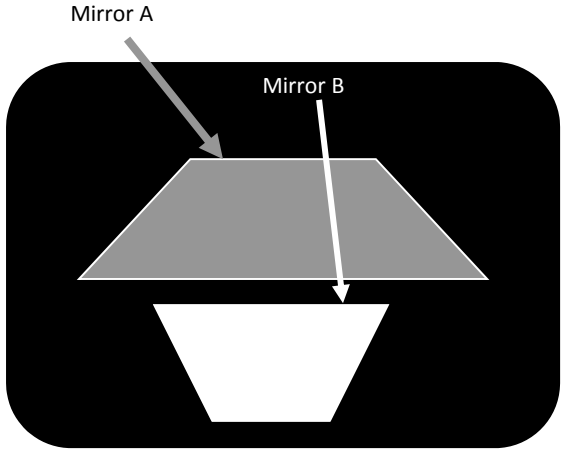
- First find the centroid of the object, and then find the furthest vertex of the object from the centroid. The distance between them will be the radius of the sphere and the centroid will be the center.



QUESTION #2

- There are two mirrors in the space and a viewer as the following configuration on the left. Assuming there is no light source nearby but only some ambient lighting. The viewer should see the picture in the screen as in the right picture.



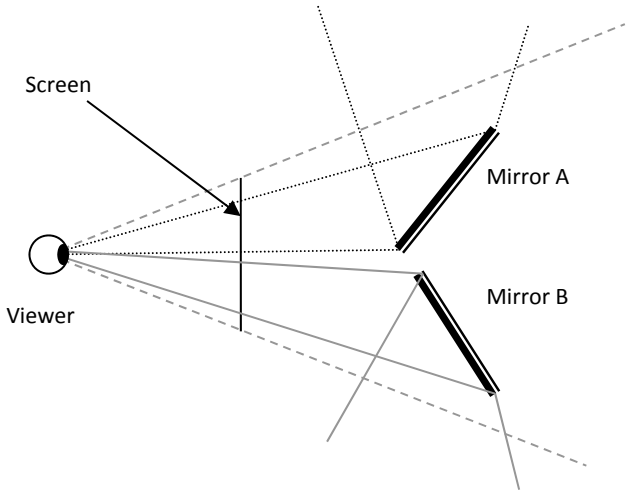


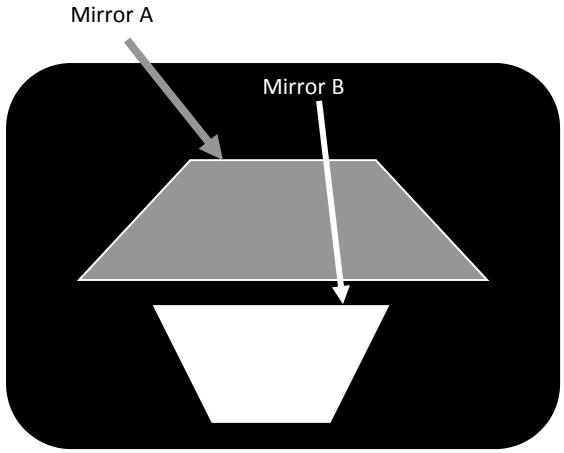
Picture to be seen on the screen

(Not drawn in the correct scale)

QUESTION #2

- If we render the picture with ray tracing, **how many times do we need to test if a ray intersects an object**? Assuming the screen dimension is 1024x768 pixels. The grey area in the picture has 100000 pixels (excluding the white area) and the white area has 70000 pixels. Assuming this is the most basic ray tracing without any other techniques to reduce the number of ray-object intersection test. However, you can assume that a ray does not intersect the object which the ray just comes out from. (Note that the viewer is NOT an object in the scene.) **List the steps and explain your answers also**.



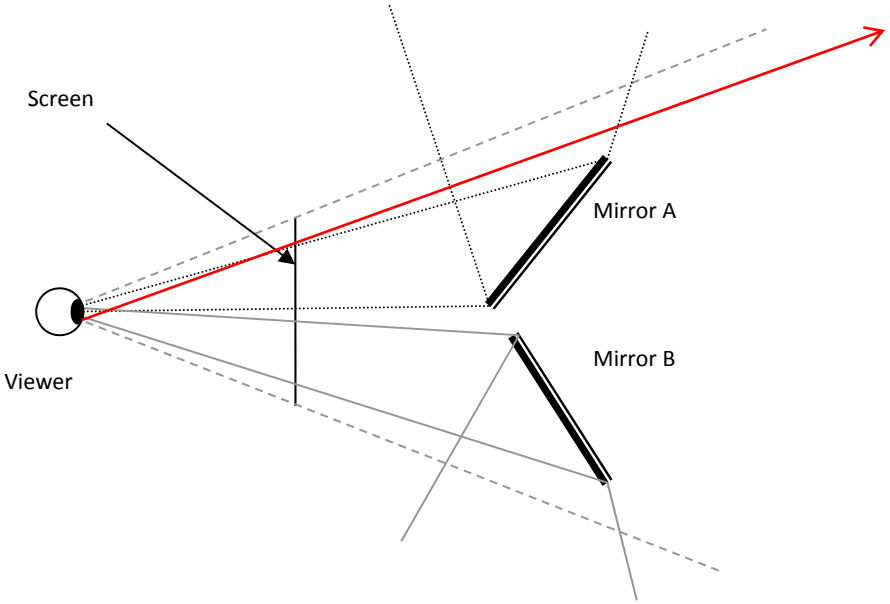


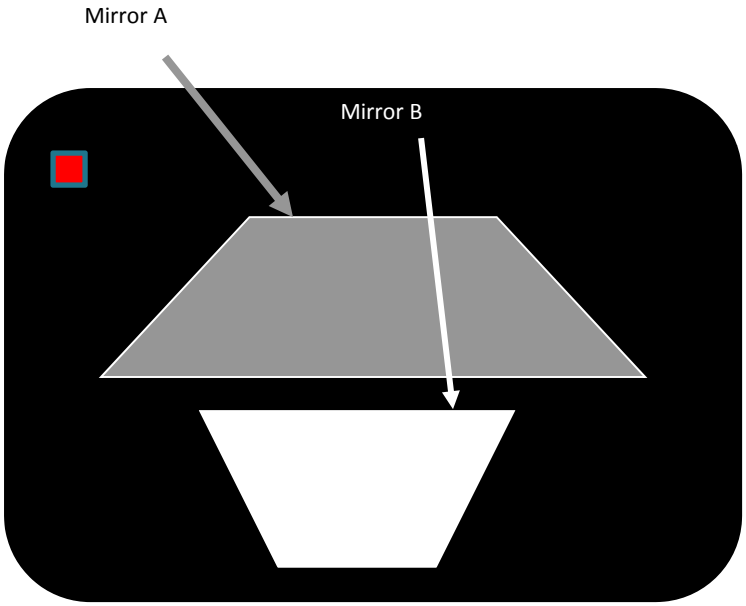
Picture to be seen on the screen

(Not drawn in the correct scale)

QUESTION #2

- 2 tests for a black pixel
  - Test against both A and B



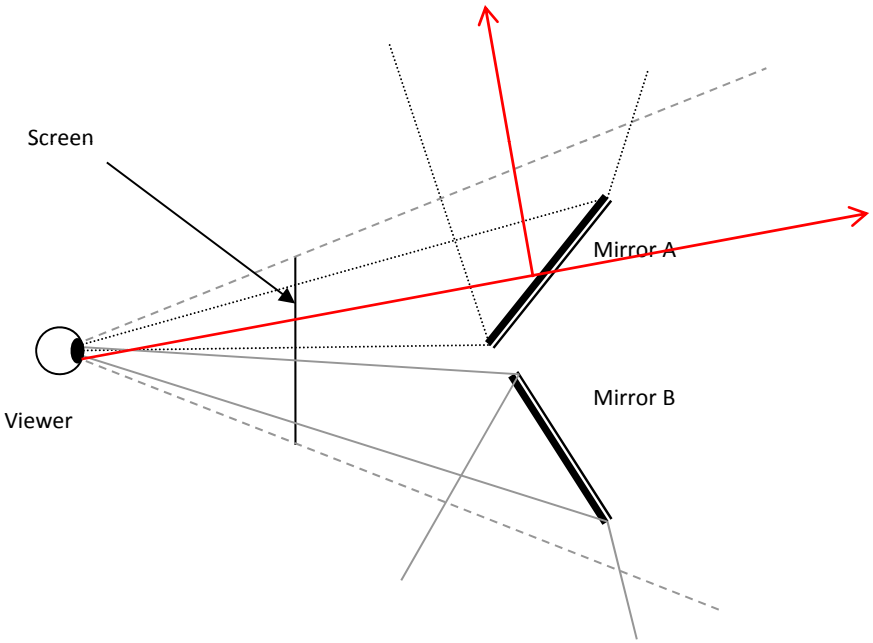


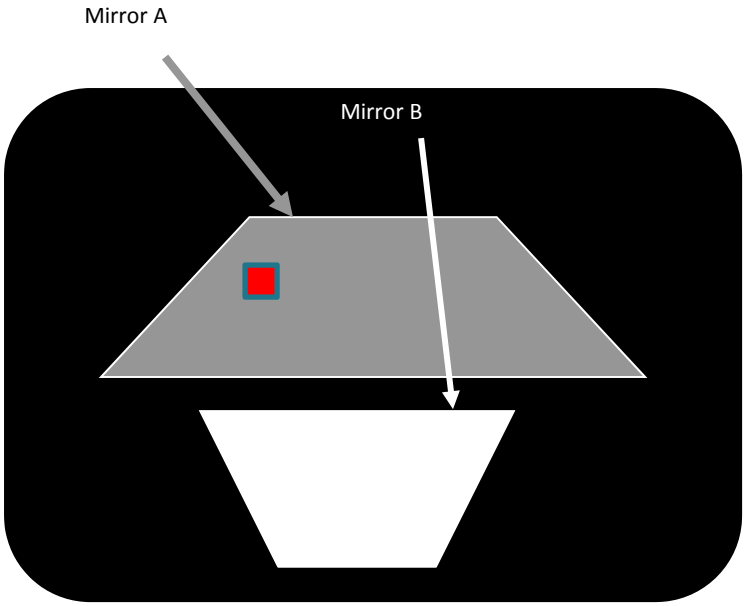
Picture to be seen on the screen

(Not drawn in the correct scale)

QUESTION #2

- 2 + 1 tests for a grey pixel





Picture to be seen on the screen

(Not drawn in the correct scale)

QUESTION #2

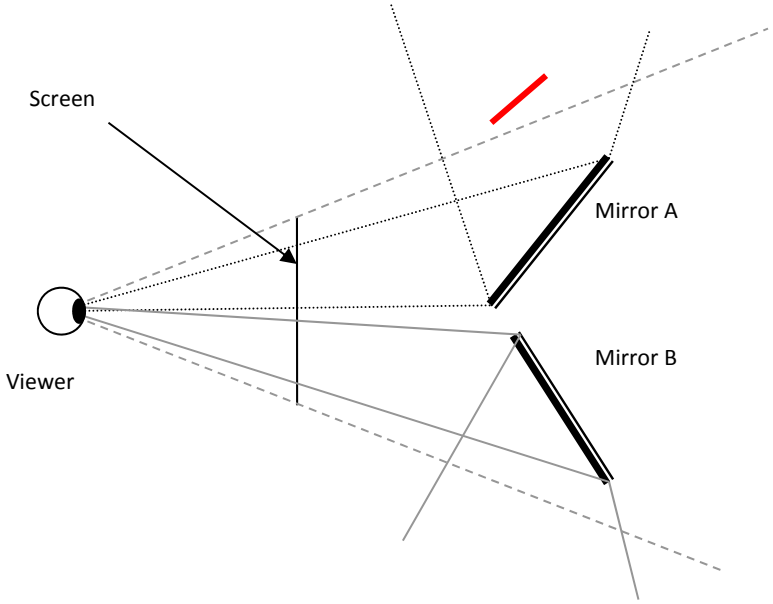
- # Black pixels = 1024 x 768 – 170000
- # Grey pixels = 100000 and # white pixels = 70000
- 2 tests on each black pixel and 3 tests on each grey or white pixel
  - Black pixel: have to test against both mirrors to make sure it doesn't hit any of them : 2 tests
  - Grey/White pixel:
    - Test against both mirrors and finds that it hits one of them : 2 tests
    - Reflected ray from mirror, have to check if it hits the other mirror - +1
- Total # of tests = 1024 x 768 x 2 + 170000

QUESTION #2

- How long will it takes to render a movie of an hour?
- Assuming the number of tests is about 1600K for each frame from last slide
- Assuming a frame rate of 30
- 1 second of movie = 30 x 1600
- 1 hour = 60 x 60 x 30 x 1600
- Assuming our computer can do 10000 tests per second
- We still need 200 days to render the movie

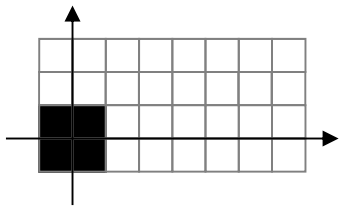
QUESTION #2

- Further more, what if we add one more mirror
- Number of tests double
- Number of tests increases exponentially with # objects

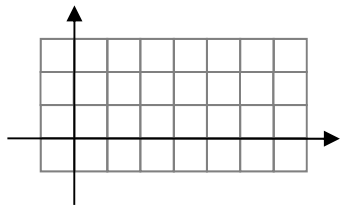


QUESTION #3

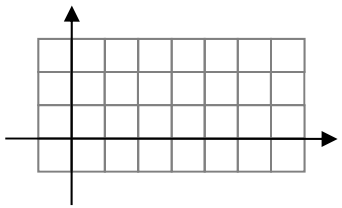
- Imagine a 2 pixels by 2 pixels square, has its center at the origin (see the left figure). The square moves linearly to a new position with its center at (1,0). If the pixels on the display are shaded as long as they are covered by the square, please draw the shaded pixels when the center of the square is at (0.5,0) and (1,0). Is the "square" a square all the time?



center at (0,0)

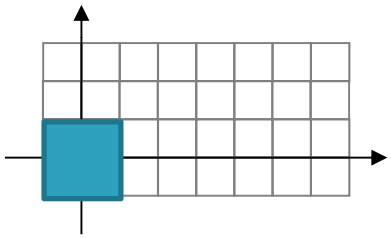


center at (0.5,0)

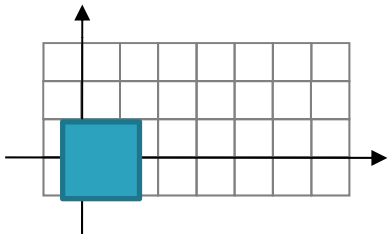


center at (1,0)

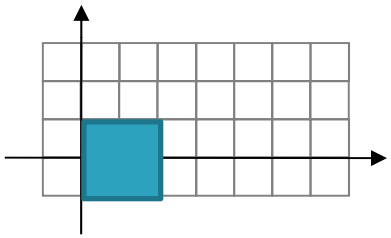
QUESTION #3



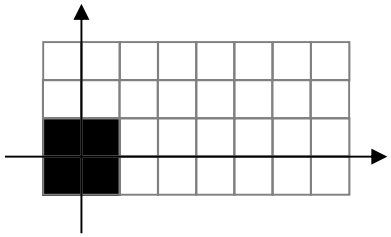
center at (0,0)



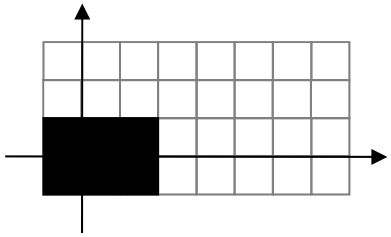
center at (0.5,0)



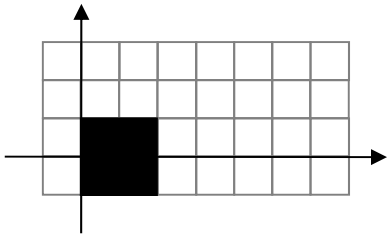
center at (1,0)



center at (0,0)



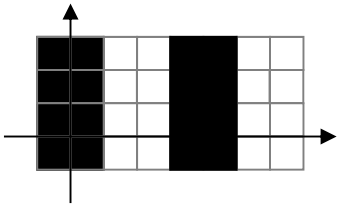
center at (0.5,0)



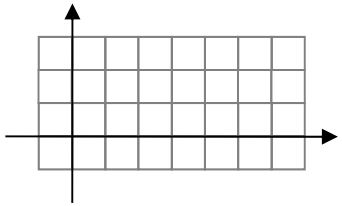
center at (1,0)

QUESTION #3

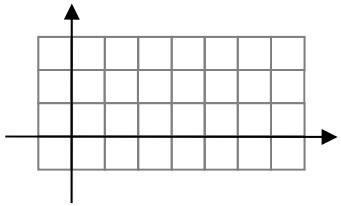
- Think about?
- The whole screen will flash



center at (0,0)

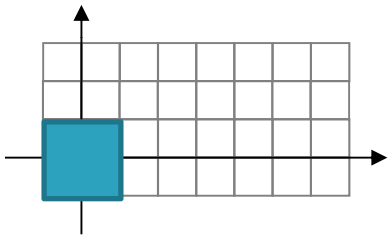


center at (0.5,0)

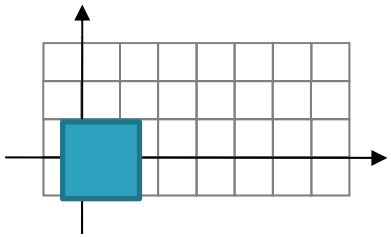


center at (1,0)

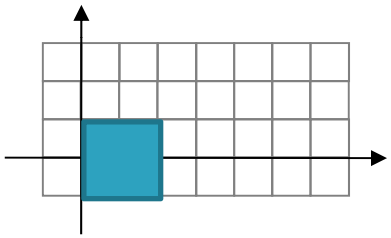
QUESTION #3



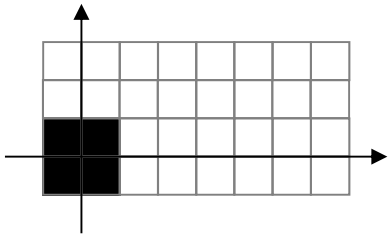
center at (0,0)



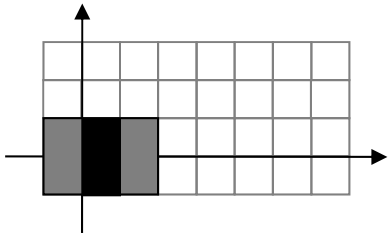
center at (0.5,0)



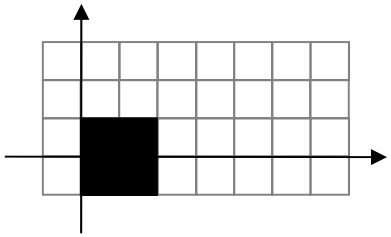
center at (1,0)



center at (0,0)



center at (0.5,0)



center at (1,0)

QUESTION #4

- In ordered-dithering, we light up a pixel if $S(x,y) > D_{ij}$. Given the dithering matrix:

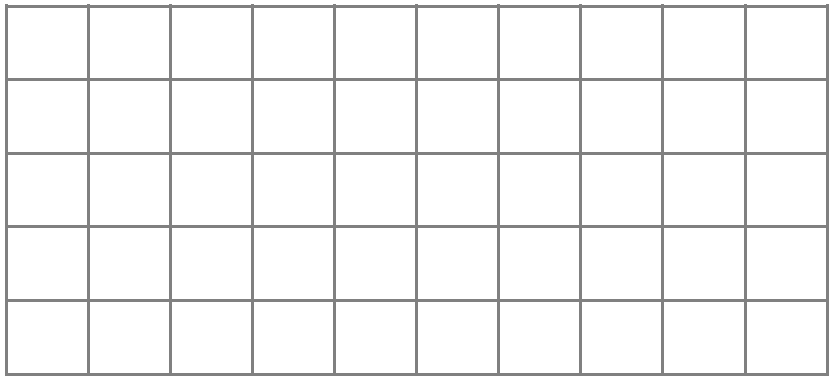
0	2
3	1

- And the pixel intensities of a picture:

4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0

QUESTION #4

- Please compute the image with ordered dither (without error diffusion). Shade the pixels which are supposed to be turned on.



- What is the average difference of pixel intensity between two pictures? Assume the intensity of the pixel is 4 when it is turned on, 0 when it is turned off.

QUESTION #4

- Overlay dithering matrix over pixel intensities

0	2
3	1

4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0

QUESTION #4

- Overlay dithering matrix over pixel intensities

0	2
3	1

4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0

QUESTION #4

- Overlay dithering matrix over pixel intensities

		0	2						
		3	1						
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0

QUESTION #4

- Overlay dithering matrix over pixel intensities

0	2
3	1

4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0

QUESTION #4

- Overlay dithering matrix over pixel intensities

				0	2				
				3	1				
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0

QUESTION #4

- Overlay dithering matrix over pixel intensities

0	2
3	1

4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0

QUESTION #4

- Overlay dithering matrix over pixel intensities

4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0

QUESTION #4

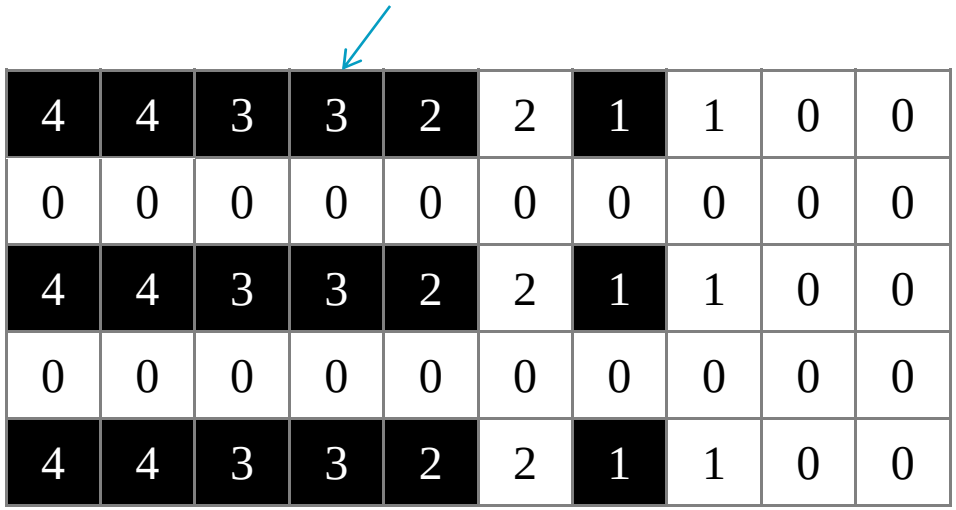
- What is the average difference of pixel intensity between two pictures? Assume the intensity of the pixel is 4 when it is turned on, 0 when it is turned off.

- Original: Avg intensity = 60/50 = 1.2
- Dither: Avg intensity = 4 x 18/50 = 1.44

4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0

THAT'S WHY WE HAVE ERROR DIFFUSION

Error = -1, distribute to the E, SE, S and SW pixels



QUESTION #4

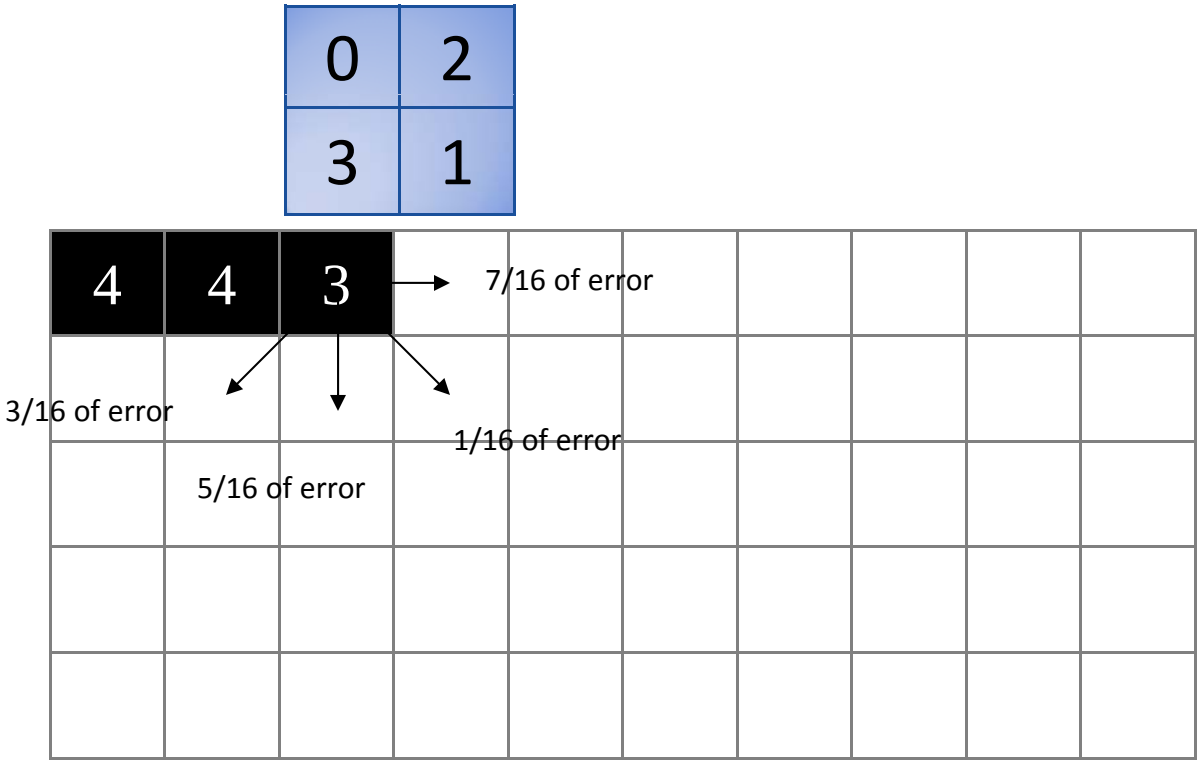
- Overlay dithering matrix over pixel intensities

0	2
3	1

4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0
0	0	0	0	0	0	0	0	0	0
4	4	3	3	2	2	1	1	0	0

QUESTION #4

- Overlay dithering matrix over pixel intensities



QUESTION #4

- Overlay dithering matrix over pixel intensities