

- Download `disk.py`
- Run the program using  
`python disk.py`
- `python disk.py -h`  
Will give all the options that you can use.
- `python disk.py -a 10 -G`  
runs the simulator in graphical mode to serve request 10 (sector number).
  - A window should appear with a simple disk on it.
  - The disk head is positioned on the outside track, halfway through sector 6.
  - The direction of rotation is anticlockwise.
  - To run the simulation, press the "s" key while the simulator window is highlighted.
  - To calculate the total time taken, you would need to know a few details about the disk. First, the rotational speed is by default set to 1 degree per time unit. Thus, to make a complete revolution, it takes 360 time units. Second, transfer begins and ends at the halfway point between sectors. Thus, to read sector 10, the transfer begins halfway between 9 and 10, and ends halfway between 10 and 11. Finally, in the default disk, there are 12 sectors per track, meaning that each sector takes up 30 degrees of the rotational space. Thus, to read a sector, it takes 30 time units (given the default speed of rotation).
  - With this information in hand, you now should be able to compute the seek, rotation, and transfer times for accessing sector 10. Because the head starts on the same track as 10, there is no seek time. Because the disk rotates at 1 degree / time unit, it takes 105 time units to get to the beginning of sector 10, halfway between 9 and 10 (note that it is exactly 90 degrees to the middle of sector 9, and another 15 to the halfway point). Finally, to transfer the sector takes 30 time units.
  - To see the answer faster without running the graphical interface, run the following  
`python disk.py -a 10 -c`
- `python disk.py -a 10,18 -G`
  - How long does this take?
  - The distance between each track is by default 40 distance units, and the default rate of seeking is 1 distance unit per unit time. Thus, a seek from the outer track to the middle track takes 40 time units.
  - The default scheduling policy is FIFO.
  - To access sector 10, which we know from above the time to be 135 time units (105 rotating, 30 transferring). Once this request is complete, the disk begins to seek to the middle track where sector 18 lies, taking 40 time units. Then the disk rotates to sector 18, and transfers it for 30 time units, thus completing the simulation. But how long does this final rotation take?

- To compute the rotational delay for 18, first figure out how long the disk would take to rotate from the end of the access to sector 10 to the beginning of the access to sector 18, assuming a zero-cost seek. As you can see from the simulator, sector 10 on the outer track is lined up with sector 22 on the middle track, and there are 7 sectors separating 22 from 18 (23, 12, 13, 14, 15, 16, and 17, as the disk spins counter-clockwise). Rotating through 7 sectors takes 210 time units (30 per sector). However, the first part of this rotation is actually spent seeking to the middle track, for 40 time units. Thus, the actual rotational delay for accessing sector 18 is 210 minus 40, or 170 time units.
- FIFO is not always best, e.g., with the request stream -a 7,30,8, what order should the requests be processed in?  

```
python disk.py -a 7,30,8 -G
```

 Run the shortest seek-time first (SSTF) scheduler (-p SSTF) on this workload.  

```
python disk.py -a 7,30,8 -p SSTF -G
```

 Notice the difference in total time taken (seek, rotation, transfer) for each request to be served.
- You should now have a basic idea of how the simulator works. Now start answering the assignment question.