1) Random seek

Consider hard disk with n tracks. Suppose the requested sectors are distributed randomly over all tracks and indipendently from each other.
Calculate the averaze number of tracks that the diskhead must cover between two requests as follows:

1) what is the prob. that the disk head is at position i and needs to move to j?

$$\frac{1}{N^2}$$

2) How many tracks need to be covered to go from i to j?

$$|i-j|$$

3) Combine ① and ② by checking all possible combinations of i and j

$$\left.\begin{aligned} \sum_{i=1}^{N} i &= \frac{1}{2}(n^2+n) \\ \sum_{i=1}^{N} i^2 &= \frac{1}{6}(2n^3+3n^2+n) \end{aligned}\right\} \text{remainder}$$

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{N^2}|i-j| &= \frac{1}{N^2}\sum_{i=1}^{N}\left(\sum_{j=1}^{i}(i-j)+\sum_{j=i+1}^{N}(j-i)\right) \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\left(i^2-\sum_{j=1}^{i}j-(N-i)i+\sum_{j=i+1}^{N}j\right) \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\left(i^2-\frac{1}{2}(i^2+i)-(N-i)i+\frac{1}{2}(N^2+N)-\frac{1}{2}(i^2+i)\right) \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\left(i^2-(N+1)i+\frac{1}{2}(N^2+N)\right) \\
&= \frac{1}{N^2}\left(\frac{1}{6}(2N^3+3N^2+N)-(N+1)\frac{1}{2}(N^2+N)+\frac{N}{2}(N^2+N)\right) \\
&= \frac{1}{N^2}(\frac{1}{3}N^3-\frac{1}{3}N)=\frac{N}{3}-\frac{1}{3N}.
\end{aligned}$$

4) Given that N is large, give an aproximation for the number of tracks to be covered, relative to the total number of tracks

$$\lim_{n\to\infty}\frac{\frac{N}{3}-\frac{1}{3N}}{N}=\lim_{N\to\infty}\frac{N^2-1}{3N^2}=\frac{1}{3}$$

2) Scheduling algorithms

shortest seek time first

IO controller with SSTF algorithm, suppose disk head is at center of the disk

1) Series of 8 requests, where the head has to pass over the midde of the disk between each one. Give the minimum (total tracks to be covered) solution to this problem
(Note tracks relative to the middle -> 512 then 511 is -1)

Solution: Start arbitrarily with disk head pointing towards positive tracks (so req. 1)
Note that the batch is viewed by SSTF not per request but as a whole. The order of the request in the batch therefore doesn't matter.
We now always ensure that the distance to the next track on the same side (compared to the center is just 1 greater than the distance to the other side
e.g. dist(1,-2)=3 then dist(1,5)=4=3+1
so for 8 req. we have

1, -2, 5, -10, 21, -42, 85, -170

2) calculate the number of track changes

1+3+7+15+31+63+127+255 = 502

3) Same but handled by LOOK algo., calculate number of track changes and cmp.
LOOK goes to a certain direction until there are no more in that dir.

0->85 -> -170 => 85+255=340 LOOK much better

RAID efficiency

Given that we have 12 40GB HDDs

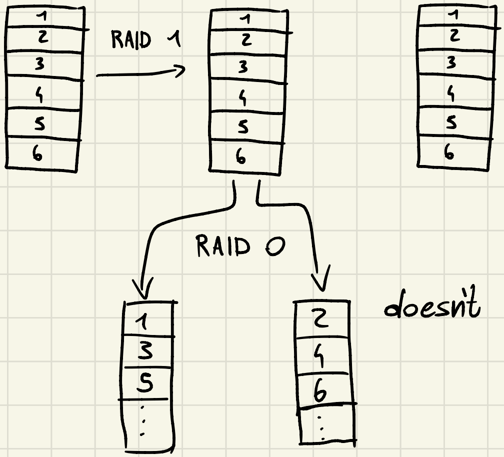
for each subsequent RAID configuration how many GBs can be placed in the system

- 0+1
- 1+0
- 0+5 } all three possible divisions
- 5+0

total GB = $12 \cdot 40 = 480$

raid 0+1

240 GB



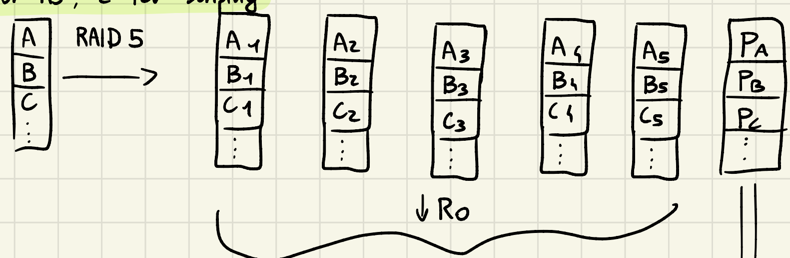
R_1 halves usable space

doesn't change available space

raid 1+0 same but it halves later 240GB

raid 0+5 first make disk for parity then spread

A) 6 for R_5, 2 for striping



I don't loose space for these

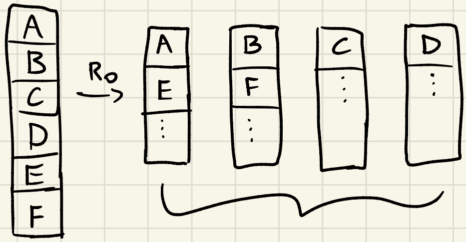
2 disks used for parity

$480 - 2 \cdot 40 = 400$GB

logic the same for the others but I increase the number of disks for RAID 0

raid 5+0

A) 3 disks for RAID5 4 for RAID 0



doesn't change anything

RAID 5

for each one a disk is used for parity

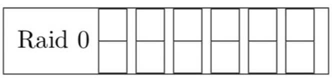
$480 - 4 \cdot 40 = 320$

As I decrease the n. of disks used for RAID 0 the available space increases (less parity disks needed

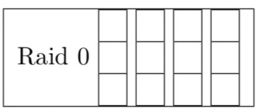
X) Raid 0+1

Y) Raid 1+0

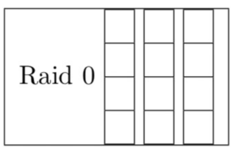
A) Raid 0+5



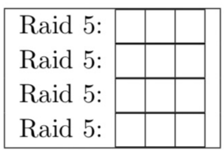
B) Raid 0+5



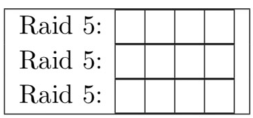
C) Raid 0+5



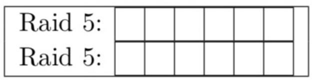
D) Raid 5+0



E) Raid 5+0



F) Raid 5+0



RAID robustness

Given we have 12 disks for the following consequential RAID configurations check the maximum number of disks that can fail without the system failing.
Then calculate the probability that the system will remain operational if two disks fail

- R 0+1
- R 1+0
- R 0+5
- R 5+0 ← all 3 config.

<u>solution</u>

the only difference is which disk fails. With all systems only 1 can certainly fail, another "wrongly selected" one will cause failure of the system.
With "well chosen" disks up to 6 disks can fail.
To determine the probability that the system will remain operational if two disks fail we can first pick a random disk that is already failing

	max n. of disks that can fail	chance operational with two disks
X)	1-6	5/1
Y)	1-6	10/11
A)	1-2	1/11
B)	1-3	2/11
C)	1-4	3/11
D)	1-4	9/11
E)	1-3	8/11
F)	1-2	6/11