**Justification**

A data structure is a collection of data type ‘values’ which are stored and organized in such a way that it allows for efficient access and modification.

In my program the data structure which I am choosing is Linear include Array and List. An array is a finite group of data, which is allocated contiguous memory locations, and each element within the array is accessed via an index key (typically numerical, and zero based).

In addition, the sorting algorithm I follow with Quick sort, it is a highly efficient sorting algorithm and is based on partitioning of array of data into smaller arrays. In efficient implementations Quick Sort is not a stable sort, meaning that the relative order of equal sort items is not preserved. Overall time complexity of Quick Sort is O(n\*Log n), the space complexity of Quick Sort is O(n\*Log n).

The better way to compare how the efficiency of different approaches to a program problem is using big O notation to analysis how quickly the runtime grows, it called "asymptotic analysis". With big O notation express the runtime in terms of—brace yourself—how quickly it grows relative to the input, as the input gets arbitrarily large. For example:

def count\_me\_up (big\_list):

for row\_list in big\_list:

do something

for column\_list in big\_list:

do something

print(row\_list,column\_list)

This is nesting two loops. If list has n items, our outer loop runs n times and Inner loop runs n times for each iteration of the outer loop, giving n^2 total prints. Thus, this function runs in O(n^2) time (or " polynomial time or square time"). If the list has 10 rows, must print 100 cells. If it has 1,000 rows, must print 1,000,000 cells.

For calculating time measurement is as follows:

import time

start\_time = time.time()

# do your work here

end\_time = time.time()

print(end\_time - start\_time)

Quick sort method, it is one of the most famous sorting algorithms based on divide and conquers strategy which results in an O(n\*log n) complexity. But, it will take more time because searching for smallest first then add smallest to new arraylist and don’t forger remove it, to make sure next time easier for comparing new value.

def Sorting(*self*,array):

Sort\_list = []

while array:

smallest = min(array)

Sort\_list.append(smallest)

array.pop(array.index(smallest))

return Sort\_list

Moreover, Because of requirement is sorting data with following index. When I append the data, I add its index follow with as well, after that the index of sorted data will store in new array then be called back and compared with the available list, if matches print off

def sort\_return\_index(self,new\_array):

return\_index = []

sort\_array=[]

for i in new\_array:

sort\_array.append([new\_array[i],i])

temp = self.Sorting(sort\_array)

for i in temp:

return\_index.append(i[1])

return return\_index

**Brexit**

The GDPR is a piece of EU legislation, but it will still matter to the UK after Brexit. A new Data Protection Bill, which will repeal the current Data Protection Act (1998), is currently making its way through the UK Parliament.

**BACKUP YOUR DATA**

Hospital systems are vulnerable to disasters — fire, flood or hurricanes for example — in addition to malicious hacking. Backing up the data on a regular basis and keeping those backups securely stored is key to recovering from disaster.

It is unlikely that an organization will be able to prevent any possible IT security incidents. That is why it is important to develop an action plan when a breach occurs.

**Secure data**

Any data of patients, passwords of staffs or users, network transmission should be encrypted to make it difficult (ideally, impossible) for hackers to interpret data especially hospital's domain website.

Keep tracking which users are accessing the data on which computer available in hospital, if not in list get id address to treat them by law. Moreover, each PCs should be installed anti-virus, keylogger application to protect and monitor staffs or users.

Create a personal identifier or other client authentication, required. This hampers ease of use, but is the first line of defence against a lost device.

**Limit old data**

When the data about an individual has served its purpose, it must be deleted or destroyed, unless there are other facilities to retain it. Organizations need a review process to clean the database.

**Provide training to physicians and hospital staff knowledge**

Employees are the important link in information security. They need to understand proper information security protocols and the consequences of not following protocols. To make sure they understand how to correctly use their remoted mobile apps, make the most of their mobile capabilities and watch for suspicious activity.

**Accuracy and consistency of data**

Ensuring the accuracy and consistency of data is an important aspect of information management. The information contained in a database must be accurate to ensure reliable results are obtained after analysis.

**Perform regular risk assessments**

Risk assessment will help understand the organization's vulnerabilities. Only then can you anticipate hospital system security and security issues properly.

**Patch electronic medical devices**

While many of the IT security threats healthcare, organizations face also affect companies in other industries, providers have another risk: the threat of pacemakers, monitoring tools and other electronic medical devices being hacked.

One step healthcare IT departments must take: Keep the software on those devices patched and up to date to minimize their vulnerabilities.

**Implement physical security controls**

Even as electronic health records become more common, organizations keep a lot of sensitive data on paper. Therefore, suppliers must ensure that doors and file cabinets are locked and that cameras and other physical security controls are used.

In addition, organizations should physically secure IT equipment by locking down the server room and using cable locks or other devices to keep laptops and desktops attached to office furniture.

**Write mobile device policies**

Because many health workers use personal devices to do their jobs, it is important for every organization to create a mobile device policy that governs what data can be stored on devices. utilities, what applications can be installed, etc.

In addition, many vendors are using mobile device management software to enforce those policies.