**Reading (VIPERS FORM)**

Use this form to record key ideas from reading to prepare for the lessons.  make sure you upload the completed form in TEAMS (Reading section)

|  |  |
| --- | --- |
| Topic you prepared: | Links used: |
| Queues | [Queues — Isaac Computer Science](https://isaaccomputerscience.org/concepts/dsa_datastruct_queue?examBoard=all&stage=all) |

Terms: Write any new technical term and their meaning, add more rows if needed.

|  |  |
| --- | --- |
| **Terms** | **Meaning** |
| enqueue(data) | Adds an element to the queue – functions |
| dequeue() | Removes an element for the queue (will remove from front) |
| **priority queue** | Is where each element in the queue has a priority. When new elements are added to the queue, they are inserted ahead of those of lower priority and behind elements of equal priority. |
| **Circular queue** | A way of implementing a queue based on a static array where free space at the start of the array is used once the end of the array is reached. |

|  |
| --- |
| **What can we understand from the topic?** |
| A queue is an abstract data type that holds an ordered, linear sequence of items. It uses the same concept as a stack however, it uses a First In, First Out (FIFO) structure. This means that the first element to be added to the queue will be the first to be removed (the same as in real life).  A queue can involve a static or dynamic implementation (can and can’t change in size).  Furthermore, There are priority queue where certain elements have priority a real life example would be Students join the queue at the end of the line of other students but prefects can join ahead of students, but behind other prefects.  The queue always removes form the front and when there is an full queue it begins from 0 again and continues the queue as seen. |

|  |
| --- |
| **How this topic linked to other previous topics ? What conclusions can we draw from this topic?** |
| It links to stacks ads they are both abstract data types that hold data in an ordered manner.  This also uses the knowledge of pointers to identify the rear and front of the queue. |

|  |
| --- |
| **What are the strengths / applications of this topic? (e.g. accuracy, steps of calculations, time or space complexity, used when …., hardware and software needs, ethical or legal issues, ….etc.)** |
| A queue can involve a static or dynamic implementation – which can configure under large amounts of data. For example, print tasks are stored on a print queue while waiting to be printed. Furthermore, if there is a set of tasks waiting to be executed, a priority queue can change the order which can be useful when wanting certain actions done. |

|  |
| --- |
| **What are the limitations / drawbacks of this this topic (e.g. problems, limited accuracy, too many steps, too complex, cannot be used when, hardware requirements, ethical and legal issues, ….. etc.)** |
| I don’t necessarily think there are any limitations beside that it could be seen as niche. |

.

|  |
| --- |
| **Write a summary of the topic in 50 words** |
| To summarise, a queue is an abstract data type that holds an ordered, linear sequence of items. It uses a FIFO structure so that when an element is added to the queue it adds to the rear and when removed removes the head. Furthermore, circle queues go into rotations of starting new arrays. |