|  |  |
| --- | --- |
| **Stack** | **Heap** |
| The stack memory segment is RAM's area that is used during program execution for storing return addresses of function calls, arguments to functions, and local variables within functions. Stack memory is maintained in Last In First Out (LIFO) fashion where new storage is allocated and deallocated at only one end, called the TOP of the stack. A special pointer called stack pointer points to the TOP of the stack.  The scope of variables allocated at stack is confined to the block--group of statements enclosed within curly braces. Once the control exits the block the C compiler pops these variables off the stack to clean up. | The heap segment is also RAM's area that is used during program execution but this is used for storing global (storage class external), and static variables. Heap is allocated variables at runtime by usually following '*first fit--choose first block that can satisfy request*' strategy.  All variables allocated in the heap remains in existence for the duration of a program. |
| Stack is a limited memory region given to a program for its execution if too much data pushed on to the stack (like large structures and arrays), it can result into a *stack overflow* and the program will abort.  Also, if a pointer is taken to something on the stack, and it is passed to a function or returned it from a function, the function receiving it will result into *segmentation fault* because the actual data will get popped off and disappear and pointer will be pointing at dead space. | To prevent *stack overflow*, large memory consuming data items should be allocated on heap on demand and must be freed when they are no longer needed. |
| Limited stack size is given to a program for its execution and it is OS-dependent. | Size of heap is not limited to a program the way stack is. A program can allocate as much memory dynamically as needed until heap exhausted. |
| Stack is very fast from access time's point of view. | Heap has slower access time than stack. |
| Data items on stack are allocated and deallocated by CPU; therefore memory will be managed efficiently, there will be no memory leaks and memory will not become fragmented. | On the other hand, for the data items allocated on heap, it's programmer's responsibility to allocated memory wisely and free it when it is no longer used by the program. If memory is not well managed there may be memory leaks and memory may become fragmented over time as blocks of memory are allocated. |
| Variables once allocated on to the stack cannot be resized. | Variables allocated from heap region can be resized using realloc() |
| Memory allocated from stack is called static memory allocation. | While, memory allocated from heap is called dynamic memory allocation. |

Priority Queue Applications :-

• Event-driven simulation. [customers in a line, colliding particles]

• Numerical computation. [reducing roundoff error]

• Data compression. [Huffman codes]

• Graph searching. [Dijkstra's algorithm, Prim's algorithm]

• Computational number theory. [sum of powers]

• Artificial intelligence. [A\* search]

• Statistics. [maintain largest M values in a sequence]

• Operating systems. [load balancing, interrupt handling]

• Discrete optimization. [bin packing, scheduling]

• Spam filtering. [Bayesian spam filter]

**Discrete event simulation[[edit](https://en.wikipedia.org/w/index.php?title=Priority_queue&action=edit&section=13" \o "Edit section: Discrete event simulation)]**

Another use of a priority queue is to manage the events in a [discrete event simulation](https://en.wikipedia.org/wiki/Discrete_event_simulation). The events are added to the queue with their simulation time used as the priority. The execution of the simulation proceeds by repeatedly pulling the top of the queue and executing the event thereon.

**Bandwidth management[**[**edit**](https://en.wikipedia.org/w/index.php?title=Priority_queue&action=edit&section=12)**]**

Priority queuing can be used to manage limited resources such as [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(computing)) on a transmission line from a [network](https://en.wikipedia.org/wiki/Computer_network) [router](https://en.wikipedia.org/wiki/Router_(computing)). In the event of outgoing [traffic](https://en.wikipedia.org/wiki/Traffic) queuing due to insufficient bandwidth, all other queues can be halted to send the traffic from the highest priority queue upon arrival. This ensures that the prioritized traffic (such as real-time traffic, e.g. an [RTP](https://en.wikipedia.org/wiki/Real-time_Transport_Protocol) stream of a [VoIP](https://en.wikipedia.org/wiki/Voice_over_Internet_Protocol) connection) is forwarded with the least delay and the least likelihood of being rejected due to a queue reaching its maximum capacity. All other traffic can be handled when the highest priority queue is empty. Another approach used is to send disproportionately more traffic from higher priority queues.

Many modern protocols for [local area networks](https://en.wikipedia.org/wiki/Local_area_network) also include the concept of priority queues at the [media access control](https://en.wikipedia.org/wiki/Media_access_control) (MAC) sub-layer to ensure that high-priority applications (such as [VoIP](https://en.wikipedia.org/wiki/VoIP) or [IPTV](https://en.wikipedia.org/wiki/IPTV)) experience lower latency than other applications which can be served with [best effort](https://en.wikipedia.org/wiki/Best_effort) service. Examples include [IEEE 802.11e](https://en.wikipedia.org/wiki/IEEE_802.11e) (an amendment to [IEEE 802.11](https://en.wikipedia.org/wiki/IEEE_802.11) which provides [quality of service](https://en.wikipedia.org/wiki/Quality_of_service)) and [ITU-T](https://en.wikipedia.org/wiki/ITU-T) [G.hn](https://en.wikipedia.org/wiki/G.hn) (a standard for high-speed [local area network](https://en.wikipedia.org/wiki/Local_area_network) using existing home wiring ([power lines](https://en.wikipedia.org/wiki/Power_line_communication), phone lines and [coaxial cables](https://en.wikipedia.org/wiki/Ethernet_over_coax)).

Usually a limitation (policer) is set to limit the bandwidth that traffic from the highest priority queue can take, in order to prevent high priority packets from choking off all other traffic. This limit is usually never reached due to high level control instances such as the [Cisco](https://en.wikipedia.org/wiki/Cisco_Systems,_Inc.) [Callmanager](https://en.wikipedia.org/w/index.php?title=Callmanager&action=edit&redlink=1), which can be programmed to inhibit calls which would exceed the programmed bandwidth limit.