Kuyruk (Queue) Veri Yapısı:

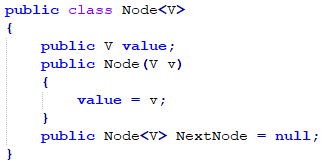
Burada veriler yine bir liste gibi tabi-i caizse tek sıra halde tutulurlar. Fakat burada kuyruğa ekleme ve çıkarma özel bir kural içerir. Kuyruğa yeni bir eleman eklendiğinde kuyruğun en sonuna eklenir ve kuyruktan bir eleman çıkarması istendiğinde, kuyruk ilk gireni çıkarır. Yani sıra önceliği vardır. Bunu da yemek kuyruğuna girmeye benzetebiliriz. İlk gelen kuyruğun en başındadır sonra gelenler sırayla kuyruğun sonuna eklenir. Yemek verilmeye başlandığında kuyruğa ilk girenden (en baştan) başlanarak kuyruktan çıkılır. Buna “ilk giren ilk çıkar” “First in Fisrt out (FIFO)” kuralı denir. Şaka değil :D

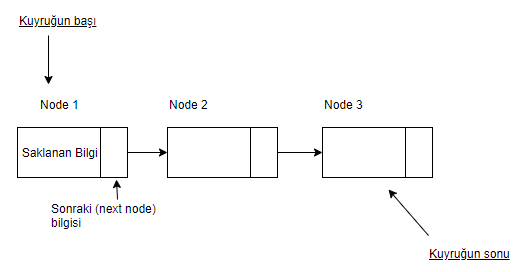
İngilizce-Türkçe yazıyorum çünkü bu kavramların ilgilizcelerini de bilmek şart.

Kuyruk (queue) yapısında temel olarak elemanların sıralı olarak tutulması gerekir. Bunun için bağlı liste (linked list) kullanılabilir. Ayrıca ekle (enqueue) ve çıkar (dequeue) methoları kuyruğun temel fonksiyonları olacaktır.

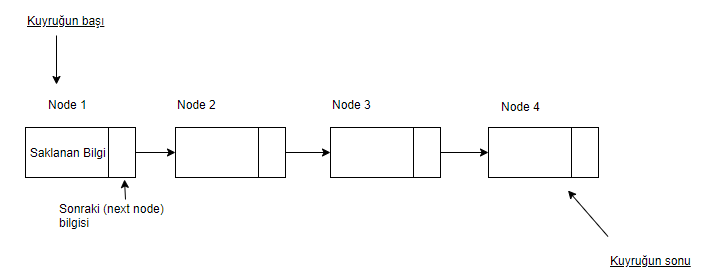
Node’umuzu yine liste node’u gibi alabiliriz:

Generic tip C# dilinde yazılmış node’umuz içinde taşıyacağı değeri ve bir sonraki elemanı göstermektedir:

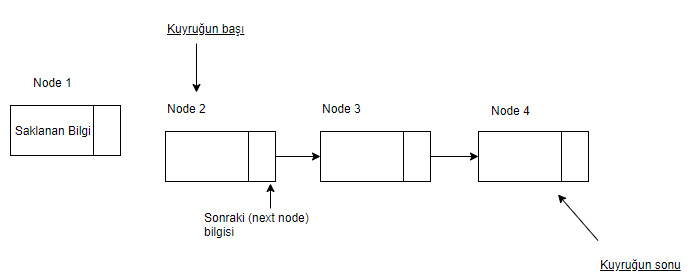




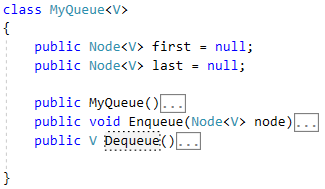
Kuyruğa yeni bir eleman eklendiğinde kuyruğun sonuna eklenir (Enqueue):



Kuyruktan br eleman çıkarması istendiğinde en baştaki eleman kuyruktan çıkarılır ve değeri döndürülür (Dequeue)



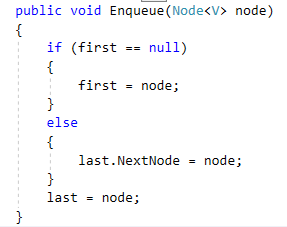
Kuyruk yapısını:



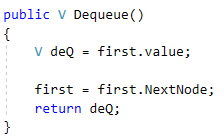
Şeklinde bir yapıya sahiptir.

Ekleme ve çıkarma methodlarını ile yine kod olarka gösterecek olursak:

Ekleme method bağlı liste gibi ilk değer başa gelecek şekilde sonrakiler de devamına gelecek şekilde eklenir:

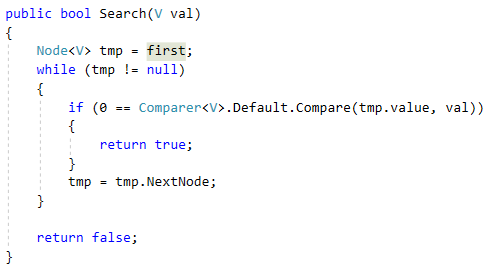


Çıkarma methody da en baitaki değerin döndürülerek bir sonrkaki nodun da ilk node olması sağlanacak şekilde ayarlanır.



Bir de “Search” (Arama) methodundan bahsetmek gerek.

Bu da o node’un kuyrukta olup olmadığı bilgisini döner. Değer kuyrukta varsa “true” yoksa “false” döner:



Zaman karmaşıklığı (Time Complexity) değerleri:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Enqueue | Dequeue | Search |
| Zaman Karmaşıklığı  (Time Complexity) | O(1) | O(1) | O(n) |

Kod’un tüm hali ve uygulamalı örneği için :

https://github.com/nilaytufek/Queue/tree/master