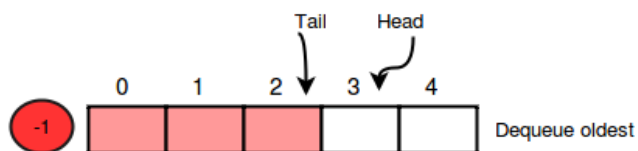
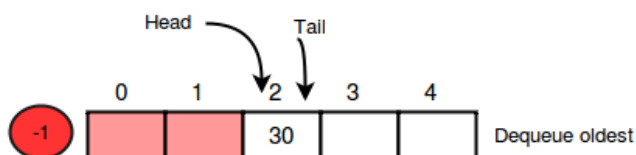
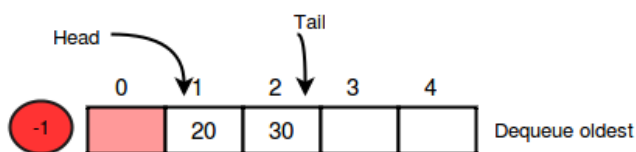
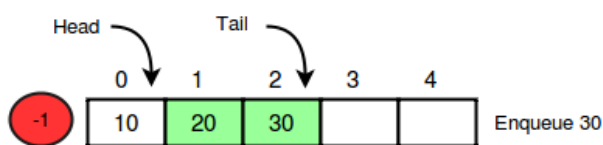
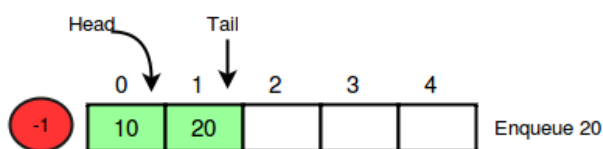
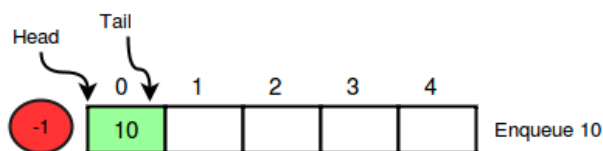
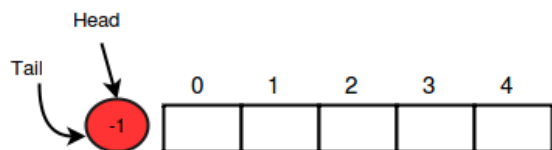


কিউ একটা বেসিক ডাটা স্ট্রাকচার। এটাকে তুমি চিন্তা করতে পারো বাসের লাইনের মত, যে সবার সামনে দাড়িয়ে আছে সে সবার আগে উঠবে, নতুন কোনো যাত্রী আসলে সে লাইনের পিছনে দাড়াবে।

কিউতে দুইরকম অপারেশন থাকে। এনকিউ(Enqueue) মানে হলো কিউতে নতুন এলিমেন্ট যোগ করা এবং ডিকিউ(Dequeue) বা পপ মানে হলো সবথেকে পুরোনো এলিমেন্টটা কিউ থেকে সরিয়ে ফেলা।

অ্যারে ব্যবহার করে আমরা ফিক্সড সাইজের কিউ ইমপ্লিমেন্ট করতে পারি। আমাদেরকে সবসময় দুইটা পয়েন্টার রাখতে হবে, হেড (Head) পয়েন্টার নির্দেশ করবে কিউয়ের সামনের এলিমেন্টের পজিশন এবং টেইল (Tail) পয়েন্টার নির্দেশ করবে পিছনের এলিমেন্টের পজিশন। একদম শুরুতে $Head = -1$, $Tail = -1$ রাখতে পারো। প্রতিটি এলিমেন্ট এনকিউ করার সময় টেইলকে এক পজিশন সামনে এগিয়ে নিয়ে সেই পজিশনে নতুন এলিমেন্টকে রাখতে হবে। ডিকিউ করার সময় শুধুমাত্র হেড একধাপ এগিয়ে নিতে হবে। একটা সিমুলেশন দেখলেই ব্যাপারটা পরিষ্কার হবে:

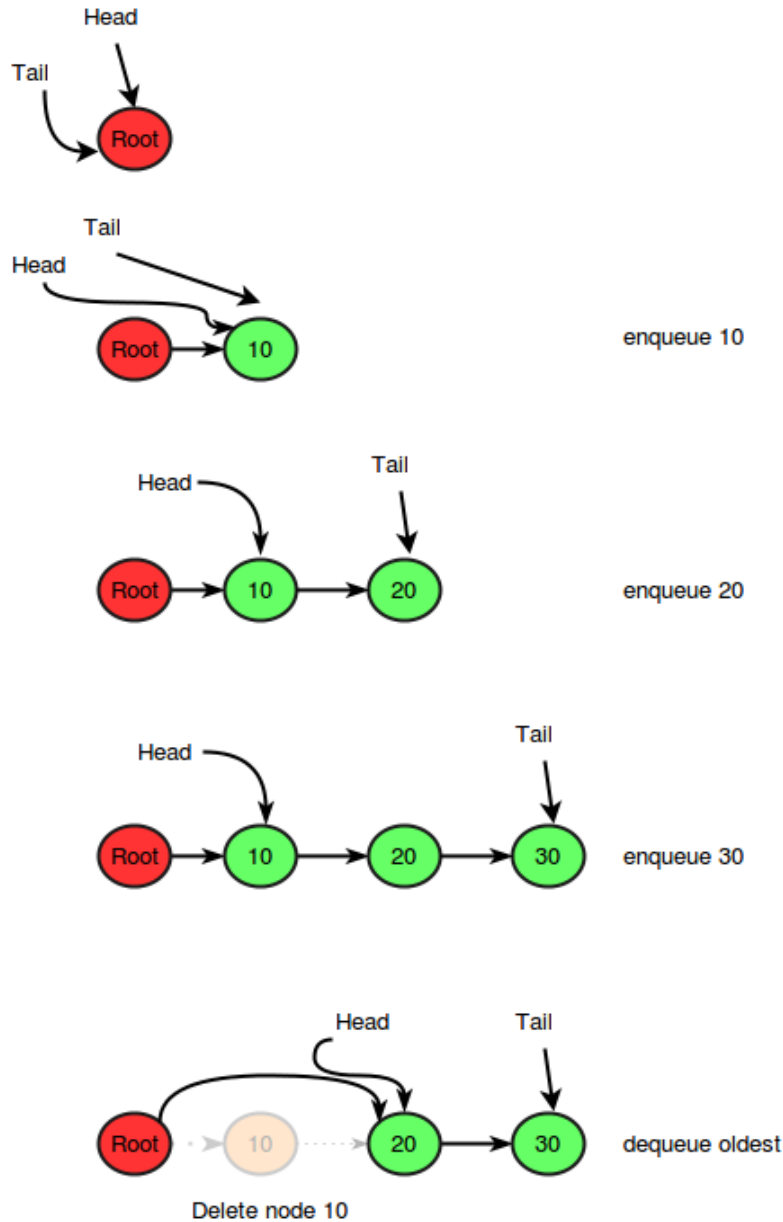


লক্ষ কর, শেষের এলিমেন্টটা ডিকিউ করার পর হেড টেইলের এর সামনে চলে গেছে। এতে কোনো সমস্যা নাই, $Head > Tail$ বা $Head = -1$ মানে হলো কিউটা পুরোপুরি খালি।

উপরের কিউতে সর্বোচ্চ ৫টা এলিমেন্ট রাখা যাবে। কিন্তু আরো বড় একটা সমস্যা আছে। Head বা Tail সবসময় সামনে আগাচ্ছে এবং ডিকিউ করার সময় অ্যারের যে জায়গাটা ফাকা হয়ে যাচ্ছে সেটা দ্বিতীয়বার ব্যবহার করার কোনো উপায় নেই! সর্বশেষ ধাপে $Tail = 2$ হয়ে গিয়েছে, যদিও

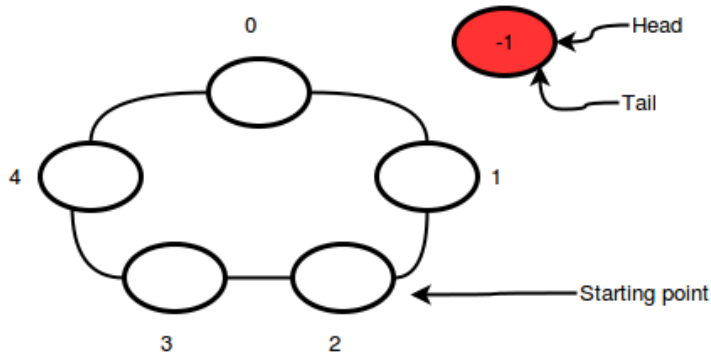
কিউটা খালি। তারমানে প্রথম ৩টা পজিশন আর ব্যবহার করতে পারবে না। এই কারণে বাস্তবে কখনোই এভাবে কিউ ইমপ্লিমেন্ট করা হয় না, করলে মেমরির সর্বোচ্চ ব্যবহার করতে পারবে না।

একটা সমাধান হলো **লিংকড লিস্ট** ব্যবহার করা। লিংকড লিস্ট ব্যবহার করলে প্রথম সুবিধা হলো কিউ এর সাইজ ফিক্সড করে দেয়া দরকার নেই। আরেকটা সুবিধা হলো তুমি যখন ডিকিউ করবে তখন টেইল পয়েন্টার যে এলিমেন্টকে পয়েন্ট করে আছে তাকে মেমরি থেকে মুছে দিতে পারবে। লিংকড লিস্ট ব্যবহার করে একটা সিমুলেশন দেখি।

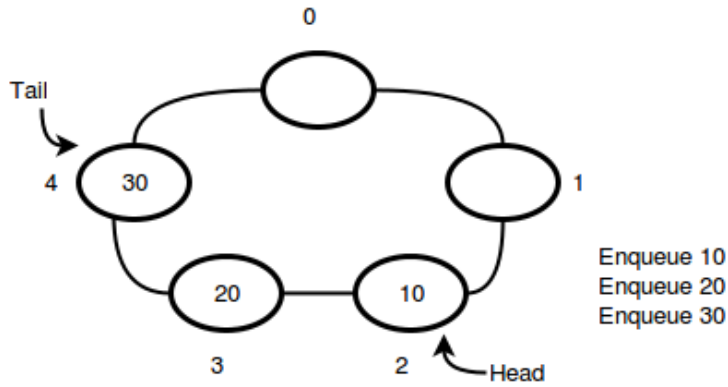


আমরা দেখতে পাচ্ছি এখানে কোন মেমরি অপচয় হবার সুযোগ নেই।

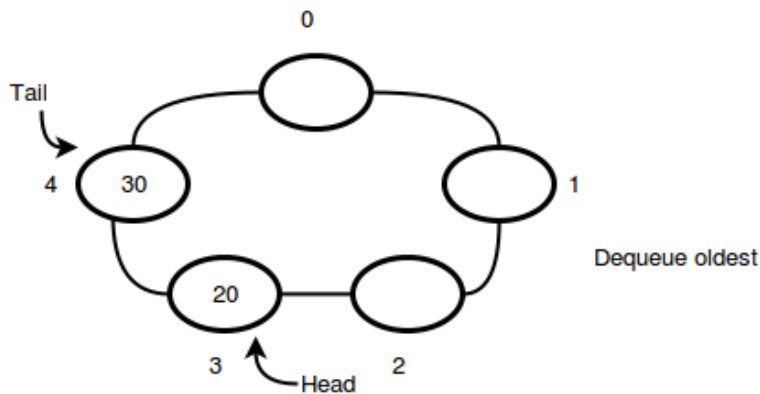
ফিক্সড সাইজের অ্যারে দিয়েও এমনভাবে কিউ ইমপ্লিমেন্ট করা যাতে মেমরি অপচয় না হয়। একে বলা হয় সার্কুলার কিউ যা দেখতে অনেকটা এরকম:



সার্কুলার কিউতে তুমি শুরুতে যেকোনো পজিশনে এনকিউ করতে পারো, উদাহরণ হিসাবে ছবিতে 2 নম্বর ইনডেক্সকে স্টার্টিং পয়েন্ট ধরেছি। আগের মতোই কিউ 10, 20 এবং 30 এনকিউ করার পর দেখতে হবে এরকম:

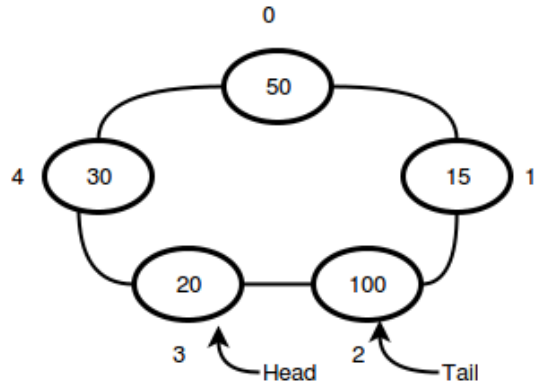


ডিকিউ ও একই ভাবে করবো:



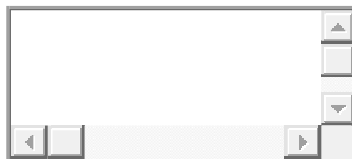
এখন মনে করো কিউটা সম্পূর্ণ ভরে

গেছে:



এখন যদি আরো একটা নতুন এলিমেন্ট এনকিউ করতে চাও তাহলে তোমার প্রয়োজনের উপর ডিপেন্ড করে দুইরকম ঘটনা ঘটতে পারে। এনকিউ ফাংশন এরোর দিতে পারে যে জায়গা খালি নেই। অথবা সবথেকে পুরানো এলিমেন্টটা ফেলে দিয়ে সেই জায়গায় নতুন এলিমেন্টটা রাখতে পারো, এটাকে বলা হয় **সার্কুলার বাফার**।

এখন প্রশ্ন হলো কিভাবে বুঝবে যে সার্কুলার কিউ ফুল হয়ে গেছে নাকি? কিউ ফুল হলে অবশ্যই টেইলের পজিশন হেডের এক ধাপ পিছনে হবে। সার্কুলার কিউ এর একটা পাইথন কোড দেখ:



```

1 Q = []
2 head = -1
3 tail = -1
4 capacity = 5
5 starting_point = 2
6 Q = [None for x in range(0, capacity)]
7 def enqueue(value):
8     global head
9     global tail
10    global capacity
11    global Q
12    global starting_point
13
14    if (tail + 1)%capacity == head:
15        print "Q is full"
16        return
17    if head == -1:
18        head = starting_point
19        tail = head
20        Q[tail] = value
21    else:
22        tail = (tail + 1)%capacity
23        Q[tail] = value
24
25 def deque():
26     global head
27     global tail
28     global capacity
29     global Q
30     global starting_point
  
```

```
31 if head == -1:
32     print "Q is already empty"
33     return
34 Q[head] = None
35 if head == tail:
36     head = -1
37     tail = -1
38 else:
39     head = (head + 1) % capacity
```

আরেক ধরনের কিউ আছে যার নাম ডাবল এন্ডেড কিউ। ডাবল এন্ডেড কিউ এর দুই পাশেই এলিমেন্ট প্রবেশ করানো যায়, আবার দুইদিক থেকেই পপ করা যায়। ডাবল এন্ডেড কিউ দিয়ে সমাধান করা যায় এমন একটা মজার সমস্যা আলোচনা করেছি [স্লাইডিং রেঞ্জ মিনিমাম কুয়েরি নিয়ে লেখায়।](#)

প্রায়োরিটি কিউ নামের আরও এক ধরনের কিউ আছে। সেখানে প্রতিটা এলিমেন্টের একটা প্রায়োরিটি থাকে, পপ করার সময় যার প্রায়োরিটি বেশি সে আগে পপ হয়। প্রায়োরিটি কিউ ইমপ্লিমেন্ট করতে হলে হিপ ডাটা স্ট্রাকচার সম্পর্কে জানতে হবে, সেটা নিয়ে আরেকদিন আলোচনা করবো।

প্রোগ্রামিং কনটেস্টে কিউ এর সবথেকে কমন ব্যবহার হলো [ব্রেথড ফার্স্ট সার্চ](#)। প্রায়োরিটি কিউ ব্যবহার করে ডায়াক্সট্রা অ্যালগোরিদম ইমপ্লিমেন্ট করা হয়। আবার অপারেটিং সিস্টেম বিভিন্ন রকমের কিউ ব্যবহার করে টাস্ক শিডিউলিং এর জন্য।

প্রতিটি বড় প্রোগ্রামিং ল্যাংগুয়েজেই কিউ লাইব্রেরি আছে যা দিয়ে খুব সহজে কিউ ব্যবহার করা যায়। কিন্তু শেখার সময় নিজেকে অবশ্যই ইম্প্লিমেন্ট করতে হবে, নাহলে কিউ কিভাবে কাজ করে বুঝতে পারবে না।

আজ এই পর্যন্তই, হ্যাপি কোডিং!