Subject: A simpler queue
Posted by luslth on Thu, 15 Sep 2016 15:26:39 GMT

Here is the typical enqueue code for a singly linked list. There is a special case when the queue is empty:

```c
if (q->head == NULL) //empty queue!
    {          
        q->head = newNode(v,NULL);
        q->tail = q->head;
    }
else
    {          
        q->tail->next = newNode(v,NULL);
        q->tail = q->tail->next;
    }
```

If we have a dummy head node (a node storing no value), then the queue starts out non-empty (i.e. the head pointer is not null and neither is the tail pointer) and never goes empty. Here is the modified constructor:

```c
q->head = newNode(NULL,NULL); //dummy head node
q->tail = q->head;
```

And here is the new enqueue:

```c
q->tail->next = newNode(v,NULL);  
q->tail = q->tail->next;
```

Can anybody tell me what dequeue would look like for a queue with a dummy head node?

---

Subject: Re: A simpler queue
Posted by rokohler on Tue, 20 Sep 2016 16:57:43 GMT

I changed my queue to use a dummy head node and came up with

```c
if(q->head->next == 0) {
    fprintf(stderr, "Dequeue from empty queue.");
    exit(-1);
}
q->size--;          
node *n = q->head->next;
q->head->next = n->next;
if(q->head->next == 0) {
    q->tail = q->head;
```
Looks good. Note that you still need a test for when the queue goes empty on dequeue. I don't know a way around that; a dummy tail node doesn't help.