







pvb1i1c:●●

draw_sprite5(size_capacity)

capcity(std::bit_cel1(capcity_))



bool try_push(const T& v)



```
size_t cur_tail = tail.load(std::memory_order_relaxed);
```

```
size_t size = current_tail - current_head;
```

```
size_t curnt_head = head.load(std::memory_order_acquire);
```

~~if (size >= (capacity - 1))~~

```
size_t index = current_tail & (capacity - 1);
```


iretvi in fadise!

```
tail.store(current_tail+1, std::memory_order_relaxed);
```

data[index] == vi;



```
std::vector<T> data{std::vector<T>(capacity)};
```



pritiivate:

```
alignas(hardware_device_interface_size) std::atomic<size_t> head{0};
```



```
alignas(hardware_device_interface_size) std::atomic<size_t> tail{0};
```



size_t capacity = 0;































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































