Homework 5

Advanced Algorithms

Problem 1:

```
#include <atomic>
template<typename T>
struct node
{
    T data;
    node* next;
    node(const T& data) : data(data), next(nullptr) {}
};

template<typename T>
class stack
{
      node<T>* old_head = head.load(std::memory_order_relaxed);
      do {
            new_node->next = old_head;
         } while(!head.compare_exchange_weak(old_head,
                                             new_node,
                                             std::memory_order_release,
                                             std::memory_order_relaxed));
      }
};
```

This code is taken from
https://en.cppreference.com/w/cpp/atomic/atomic/compare_exchange.
Ignoring the memory_order parameters, explain in your own words how this code works, using scenarios where two threads are trying to push.

Problem 2:

In the linked list data structure with fine grained locking, the remove operation needs to lock the victim node (Slide 22, LinkedList.pdf). Show how a concurrent insert after the victim creates havoc. Your solution should have lots of images, showing the status after each step. If you are using Windows and have Powerpoint installed, this would be a good way to generate images.