Thread Pool

You are tasked with constructing a two different thread pool job queue types. A thread pool is a way of pre-allocating threads and assigning them tasks at runtime. If there isn't any job to work on, the thread will wait until a job has been allocated to it.

```
struct thread_job;
struct thread_queue;
/**
* Constructs a thread queue object
*/
struct thread_queue* thread_queue_new();
/**
* Destroys the thread queue
*/
void thread_queue_destroy(struct thread_queue* q);
/**
* Enqueues a new job to the thread pool
*/
void thread_queue_enqueue(struct thread_queue* q, struct thread_job* job);
/**
* Removes a job from the thread queue
*/
struct thread_job* thread_queue_deque(struct thread_queue* q);
/**
* Checks to see if the thread queue is empty.
*/
int thread_queue_empty(struct thread_queue* q);
```

Use appropriate synchronisation primitives to ensure thread-safe communication between your main thread and the thread pool.

Thread Pool Discussion

- What is similar between a thread pool and OpenMP?
- What is different between a thread pool and OpenMP?
- Could we change how our thread pool operates?
- Outline what your scheduling method is and discuss if there is other scheduling methods with your instructor.

DNA Complement

DNA replication is the process of producing two identical DNA replicas from the original strand of DNA. It is fundamental for cell division and, hence for human life, growth and recovery. The basic principle behind DNA replication is DNA base complementarity.

Use the following table and testing appliaction available at Harvard Reverse and/or complement DNA sequences. After solving the problem, attempt use OpenMP to parallel the computation.

You may use the website above and the test data on Ed.

Run-Length Encoding and Decoding

You will need to construct a program that will encode and decode using run length encoding. Given a string in the following form.

```
AAGGHHHHHHTTTT
```

Your program would output the encoded string.

```
2A2G6H4T
```

Make the assumption that your program will only handle values within the alphabet (A-Z). Construct a solution with that assumption.

- What are the limitations with encoding with just A-Z?
- Could we decise a solution that will allow us to encode any data?
- How could we use OpenMP to distribute the workload amongst multiple threads?

Consider encoding half-bytes and map that to an alphabet.