

Tech Internship Evaluation

Instructions

Please read the instructions carefully and take the evaluations.

You have 240 minutes to finish this evaluation. Please attempt all the problems, and try not to skip any problems completely. You can use any online resource, but cannot ask for help from any person. After 240 minutes, please submit your solution to careers@jqinvestments.com. Include all source code. Everything you write should be in English.

Please try all the four evaluations as you can. Everything of your writing and solution will be counted for scores. The goal for the evaluation is to score maximum points. Please make sure that you are efficient with your time for every questions.

After you have submitted your solution, please spend 30 minutes writing a self-evaluation (in English). What did you think of the problems, how do you think you did and how do you decide to allocate your time and why? Send the self-evaluation to the above email as well.

Problem I (10 Points)

Part A. .

There are two loops which will increase the values in a 2-value array. Please read the code below and answer the question.

```
int * a = new int[2];
int steps = 256 * 1024 * 1024;
// loop A
for (int i=0; i<steps; i++) { a[0]++; a[1]++; }
// loop B
for (int i=0; i<steps; i++) { a[0]++; a[0]++; }
```

Question:

Between loop A and loop B, which one is faster? Please give your reasons. (4 points)

For example:

I think Loop x is faster, because xxx...

Part B.

Suppose we have function “updateCounter”, it will accept an input parameter named as “position”. Now we call this function by two ways.

```
int *counter = new int[1024];
void updateCounter(int position)
{
    for (int j = 0; j < 100000000; j++)
    {
        counter[position] = counter[position] + 8;
    }
}
```

Situation 1

Invoke updateCounter with parameters(position) 0,1,2,3 from four different threads, and wait until all threads are finished.

Situation 2

Invoke updateCounter with parameters(position) 16,32,48,64 from four different threads, and wait until all threads are finished.

Question:

Between situation 1 and situation 2, which one will be faster? Please give your reasons. (6 points)

For example:

I think Situation x is faster, because xxx...

Please put your answers in a separated file named as answer_1. Your answer should be in English with at least 3 sentences for each question.

Problem II (10 Points)

Suppose we have to take N orders in a row for one stock, the i th order costs $M[i]$ money. We have K accounts in total. The trader's strategy is simple, take as many orders as possible before running out of money, then switch to the next account. At the beginning, we can set equal amount of money to all K accounts.

Question:

What's the minimum amount of money to set so that trader can finish the task?

Example:

$N = 10$

$M = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

$K = 4$

In this case, the answer is 17. Trader can use the first account to take $\{1, 2, 3, 4, 5\}$, then switch to the second account and take $\{6, 7\}$, then 3rd account to take $\{8, 9\}$, and the last account to take $\{10\}$. The money for each account should be at least 17, otherwise he can not finish with 4 accounts.

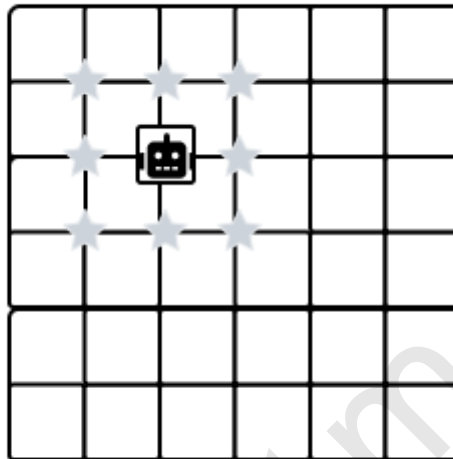
For simplicity, you can assume $M[i]$ are all integers.

You can refer to the file “problem2.cpp”/“problem2.py” in “codes” folder as an example for input/output format.

Problem III (10 Points)

Suppose there is a JQ robot in a 2D plane. Its initial position is $(0, 0)$. Each second, it can go 1 step in 8 directions.

For example, if the robot at $(2,2)$ at second i , it can move to $(1,1)$ $(1,2)$ $(1,3)$ $(2,1)$ $(2,3)$ $(3,1)$ $(3,2)$ $(3,3)$ at second $i+1$.



Questions:

1. What's the expectation of its distance to $(0, 0)$ after n seconds? Notice that the distance here means Euclid distance. Please submit answers for $n = [1, 20]$ in a separated txt file named as answer_3A. Keep precision of $1e-3$ is enough. You can refer to the file "problem3.cpp"/"problem3.py" in "codes" folder as an example for input/output format. (6 points)
2. Can you give a general formula of n ? If not, what's the biggest n can your programme calculate? Please briefly explain the reasons in a separated txt file named as answer_3B. (4 points)

All your answers should be in English and the reason for 3B should have at 3 sentences.

Example:

If $n = 1$, there are 8 possible locations for the robot. Among them, 4 are distance 1 to $(0, 0)$ and 4 are 1.414. So the expectation should be 1.207.

Notice that for this problem, you are also allowed to use mathematical methods to solve it instead of programming if you like.

Problem IV (10 Points)

Concurrent programming comes in many styles, but a particularly simple one is message passing, where threads or actors communicate by sending each other messages. One of the proponents, Golang, says in its official document:

Do not communicate by sharing memory; instead, share memory by communicating.

Golang has `chan` which is quite handy for communicating across threads, it is essentially a MPMC (multiple producer multiple consumer) queue. Unfortunately, C++ does not provide such a type.

We provide you the interface of `JqChannel`, please implement your own version of "channel" in cpp (7 points) and answer the questions below. If you use any online resources, please link them to your answer. Your code will be evaluated in correctness first and then speed. You can refer to the file "problem4.cpp" in "codes" folder as a naive example.

```
template <typename T>
class JqChannel
{
public:
    explicit JqChannel(uint32_t capacity) : capacity_(capacity) {}
    JqChannel(const JqChannel &) = delete;
    JqChannel &operator=(const JqChannel &) = delete;

    void send(const T &t);
    void send(T &&t);
    T receive();

private:
    uint32_t capacity_;
    // your code here...
};

// Your Implementation //
```

Questions:

1. How many mutexes and condition variables do you need? Why? For each mutex or condition variable, please explain its intention in 1 sentence. (2 points)
2. Can you list some concurrency scenarios where "channel" is not the best choice? Please elaborate on the reason in 1 sentence. (1 point)

Please put your answers for these two questions in a separated txt file named as `answer_4` in English.

Tips:

1. When the queued message count is larger than capacity, the sender will block until receiver consumes the message. Note that it shall support 0 capacity like Golang's `chan`.
2. To simplify the problem, `JqChannel` only needs to support `int` as message type.