CSCI 3333 Practice Quiz GRPH1

For the problems here, use the following Vertex class declaration.

class Vertex
{
    public:
        unordered_set<Vertex*> neighs;
};

Problem 1. Complete the following implementation of a function that returns whether the directed graph with vertex set $V$ contains a a cycle of length 2.

bool loopdeloop(unordered_set<Node*> &V)
{
    for (Node* v : V)
    {
        for (Node* vn : __________->neighs)
        {
            if (vn == __________)
            {
                return __________;
            }
        }
    }
    return __________;
}

Problem 2. Fill in the blanks about the function loopdeloop in Problem 1.

The worst-case running time is $\Theta($ function of $|V|$ $)$.

The best-case running time is $\Theta($ function of $|V|$ $)$.

The worst-case running time is $\Theta($ function of $|V|,|E|$ $)$.

The best-case running time is $\Theta($ function of $|V|,|E|$ $)$.
**Problem 3.** Complete the following implementation of a function that returns whether an undirected graph is 3-regular: all vertices have degree 3.

```cpp
bool threeregular(unordered_set<Node*> &V)
{
    for (Node* v : V)
        if (v->neighs.size() _____ _____)
            return _____;
    return _____;
}
```

**Problem 4.** Fill in the blanks about the function `threeregular` in Problem 3.

The worst-case running time is $\Theta(______)$.

```latex
\text{function of } |V|
```

The best-case running time is $\Theta(______)$.

```latex
\text{function of } |V|
```

The worst-case running time is $\Theta(______)$.

```latex
\text{function of } |V|,|E|
```

The best-case running time is $\Theta(______)$.

```latex
\text{function of } |V|,|E|
```
Problem 5. Complete the following implementation of a function that returns whether the directed graph with vertex set $V$ contains any two vertices with a common neighbor (i.e., edges $(v_i, v_k)$ and $(v_j, v_k)$ for three distinct vertices $v_i, v_j, v_k$).

```cpp
bool common(unordered_set<Node*> &V)
{
    for (Node* vi : V)
        for (Node* vj : V)
        {
            if (vi == vj)
                continue;

            for (Node* vk : vi->neighs)
                if (_____->neighs.find(vk) != vj.end())
                    return _____;
        }

    return _____;
}
```

Problem 6. Fill in the blanks about the function common in Problem 5.

The worst-case running time is $\Theta(\text{function of } |V|)$.

The best-case running time is $\Theta(\text{function of } |V|)$.

The worst-case running time is $\Theta(\text{function of } |V|, |E|)$.

The best-case running time is $\Theta(\text{function of } |V|, |E|)$. 

3