Floyd Warshall Algorithm

\( N \) is the number of vertices in a directed graph \( G \). The Floyd Warshall algorithm solves the all pairs shortest path problem. The name of each vertex is an integer in the range 0…\( N - 1 \).

We let \( W[i][j] \) be the weight of the arc from \( i \) to \( j \). We let the default value of the array be “infinity,” that is, we say \( W[i,j] = \infty \) if there is no arc from \( i \) to \( j \). C++ actually has no variable called “infinity.” How do we handle that? Here is C++ code.

```cpp
int V[N][N];
int back[N][N];

int main()
{
    for(int i = 0; i < N; i++)
        for(int j = 0; j < N; j++)
        {
            V[i][j] = W[i][j];
            back[i][j] = i;
        }
    for(int i = 0; i < N; i++)
        V[i][i] = 0;
    for(int j = 0; j < N; j++)
        for(int i = 0; i < N; i++)
            for(int k = 0; k < N; k++)
            {
                int temp = V[i][j]+V[j][k];
                if(temp < V[i][k])
                {
                    V[i][k] = temp;
                    back[i][k] = back[j][k];
                }
            }
    return 1;
}
```

Here is an example for \( N = 5 \). The time complexity of this algorithm is \( \Theta(N^3) \). What is the loop invariant of the outer loop? That loop invariant is important for proving the correctness of the algorithm.