Considere uma enumeração qualquer $1, 2, \ldots, |V|$ dos vértices de G. A matriz de adjacências $G.A$ de dimensão $|V| \times |V|$ é dada por:

$$G.A[i][j] = \begin{cases} 1, & \text{se } (i,j) \in G.E \\ 0, & \text{caso contrário.} \end{cases} \tag{1}$$

O pseudocódigo a seguir apresenta o algoritmo BFS onde o grafo G é representado por sua matriz de adjacências:

```
1  for i = 1 to |V| do
2      i.color ← WHITE;
3  end
4  s.color ← GRAY;
5  Q ← ∅;
6  enqueue(Q, s);
7  while Q ≠ ∅ do
8      u ← dequeue(Q);
9      for i = 1 to |V| do
10         if G.A[u][i] = 1 and i.color = WHITE then
11             i.color ← GRAY;
12             enqueue(Q, i);
13         end
14     end
15 end
```

Algoritmo 1: BFS(G, s)

Qual é a complexidade de tempo de BFS neste caso?

Solução. O laço das linhas 1-3 é executado $|V|$ vezes, enquanto que as operações das linhas 4, 5 e 6 são executadas em tempo constante. O laço das linhas 7-15 é executado enquanto a fila Q possuir algum elemento. A fila é iniciada com o vértice s que é removido da fila na linha 8, quando o laço FOR (linhas 9-14) percorre toda a linha da matriz de adjacências $G.A$ correspondente ao vértice s, ou seja, é executado $|V|$ vezes. Cada vértice desta linha, ou seja, cada vértice adjacente a s é marcado como visitado (linha 11) e inserido na fila Q (linha 12). Em seguida o laço FOR é executado novamente para o vértice que foi removido da fila. Note que cada vértice é inserido na fila uma única vez depois que ele é marcado como visitado (GRAY), e portanto o laço FOR é executado uma vez para cada um dos $|V|$ vértices visitados. Como cada execução percorre toda a linha da matriz, o custo total do laço WHILE é quadrático. Assim, podemos concluir que a complexidade de BFS neste caso é $O(V^2)$.