1 Pseudokod

```
Data: Graph G, Vertice s, Adjacency_matrix m
for each vertice u in G do
    dist[u] = infinity;
    u.prev = null;  // Iterujemy po wszystkich wierchołkach grafu, więc
     złożoność przynajmniej O(|V|)
end
dist[s] = 0;
s.prev = s;
Q = [s];            // Kolejka Q do przechowywania kolejnych wierchołków
while Q is not empty do
    u = Q.eject();
    for i = 0; i < |V|; i++ do
        if Adjacency check in adjacency matrix then
            if dist[v_i] == infinity then
                Q.inject(v_i);
                dist[v_i] = dist[u] + 1;
                v_i.prev = u;       // Do sprawdzania, które wierzchołki są
                 sąsiadujące musimy przejrzeć cały rząd macierzy,
                 dodatkowo w najtrudniejszym przypadku będziemy
                 musieli przejść przez wszystkie wierzchołki grafu w pętli
                 while co da nam złożoność obliczeniową O(|V|^2)
            end
        end
    end
end
```

Algorithm 1: BFS(Graph G, Vertice s, Adjacency_matrix m)

2 Opis

Algorytm BFS przy użyciu macierzy sąsiedztwa ma złożoność obliczeniową $O(|V|^2)$, która różni się od tej w przypadku użycia listy sąsiedztwa (O(|V| + |E|)).