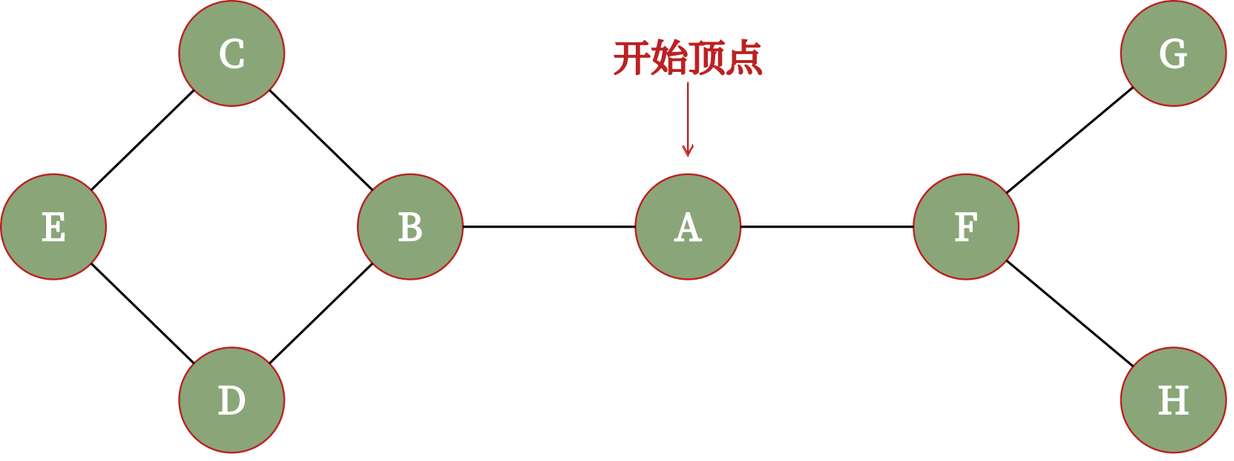

Input: 图 $G = (V, E)$，起始顶点 s
Output: 广度优先遍历的访问序列

广度优先遍历非递归算法

```
1  创建一个空的集合 visited 用于存储已访问的顶点;
2  创建一个空的队列 queue 并将起始顶点 s 入队;
3  将 s 标记为已访问，即加入到 visited;
4  while queue 非空 do
5      弹出队首元素 v;
6      输出顶点 v;
7      for 邻接顶点 w of v do
8          if w 不在 visited 中 then
9              将 w 标记为已访问，即加入到 visited;
10             将 w 入队 queue;
11         end
12     end
13 end
```


初始状态:

visited = []
queue = ['A']
将'A'加入到 visited， visited= ['A']

第一次 while 循环

visited = ['A', 'B', 'F']
queue = ['B','F']

从队列中弹出 'A'
第一次 for 循环：
检查'A'的邻接顶点'B'，由于'B'不在visited中，将其加入到visited中，visited = ['A', 'B'],同时将'B'入队，queue = ['B'].
第二次 for 循环：
检查'A'的邻接顶点'F'，将其加入到visited中，visited = ['A', 'B', 'F'],同时将'F'入队，queue = ['B','F'].

第二次 while 循环

visited = ['A', 'B', 'F', 'C', 'D']
queue = ['F', 'C', 'D']

从队列中弹出 'B'
第一次 for 循环：
检查'B'的邻接顶点'C'，由于'C'不在visited中，将其加入到visisted中，visited = ['A', 'B', 'F', 'C']，同时将其入队，queue = ['F', 'C'].
第二次 for 循环：
检查'B'的邻接顶点'D'，由于'D'不在visited中，将其加入到visisted中，visited = ['A', 'B', 'F', 'C', 'D']，同时将其入队，queue = ['F', 'C', 'D']

第三次 while 循环

visited = ['A', 'B', 'F', 'C', 'D', 'G', 'H']
queue = ['C', 'D', 'G', 'H']

从队列中弹出 'F'
第一次 for 循环：
检查'F'的邻接顶点'G'，由于'G'不在visited中，将其加入到visisted中，visited = ['A', 'B', 'F', 'C', 'D', 'G']，同时将其入队，queue = ['C', 'D', 'G']
第二次 for 循环：
检查'F'的邻接顶点'H'，由于'H'在visited中，将其加入到visisted中，visited = ['A', 'B', 'F', 'C', 'D', 'G', 'H']，同时将其入队，queue = ['C', 'D', 'G', 'H']

第四次 while 循环

visited = ['A', 'B', 'F', 'C', 'D', 'G', 'H', 'E']
queue = ['D', 'G', 'H', 'E']

从队列中弹出 'C'
第一次 for 循环：
检查'C'的邻接顶点'B'，由于'B'在visited中，停止
第二次 for 循环：
检查'C'的邻接顶点'E'，由于'E'不在visited中，将其加入到visisted中，visited = ['A', 'B', 'F', 'C', 'D', 'G', 'H', 'E']，同时将其入队，queue = ['D', 'G', 'H', 'E']

第五次 while 循环

visited = ['A', 'B', 'F', 'C', 'D', 'G', 'H', 'E']
queue = ['G', 'H', 'E']

从队列中弹出 'D'
第一次 for 循环：
检查'D'的邻接顶点'B'，由于'B'在visited中，停止
第二次 for 循环：
检查'D'的邻接顶点'E'，由于'E'在visited中，停止。

第六次 while 循环

visited = ['A', 'B', 'F', 'C', 'D', 'G', 'H', 'E']
queue = ['H', 'E']

从队列中弹出 'G'
第一次 for 循环：
检查'G'的邻接顶点'F'，由于'F'在visited中，停止

第七次 while 循环

visited = ['A', 'B', 'F', 'C', 'D', 'G', 'H', 'E']
queue = ['E']

从队列中弹出 'H'
第一次 for 循环：
检查'H'的邻接顶点'F'，由于'F'在visited中，停止

第八次 while 循环

visited = ['A', 'B', 'F', 'C', 'D', 'G', 'H', 'E']
queue = []

从队列中弹出 'E'
第一次 for 循环：
检查'E'的邻接顶点'C'，由于'C'在visited中，停止.
第二次 for 循环：
检查'E'的邻接顶点'D'，由于'D'在visited中，停止。

由于队列已空，停止循环，广度优先遍历的序列为：['A', 'B', 'F', 'C', 'D', 'G', 'H', 'E']