

# 广度优先搜索(BFS)

下面是利用队列的BFS非递归实现

```
void bfs(int v) //以v开始做广度优先搜索 (非递归实现, 借助队列)
{
    list<int>::iterator it;
    visited[v] = true;
    cout << v << " ";
    queue<int> myque;
    myque.push(v);
    while (!myque.empty())
    {
        v = myque.front();
        myque.pop();
        for (it = graph[v].begin(); it != graph[v].end(); it++)
        {
            if (!visited[*it])
            {
                cout << *it << " ";
                myque.push(*it);
                visited[*it] = true; //访问过
            }
        }
    }
    cout << endl;
}
```

From:  
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:%E5%B9%BF%E5%BA%A6%E4%BC%98%E5%85%88%E6%90%9C%E7%B4%A2\\_bfs\\_%E4%B8%BE%E6%A0%87%E6%95%B0%E6%9C%80%E7%9F%AD%E8%B7%AF\\_dijkstra&rev=1589794768](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:%E5%B9%BF%E5%BA%A6%E4%BC%98%E5%85%88%E6%90%9C%E7%B4%A2_bfs_%E4%B8%BE%E6%A0%87%E6%95%B0%E6%9C%80%E7%9F%AD%E8%B7%AF_dijkstra&rev=1589794768)

Last update: 2020/05/18 17:39