



Simple Code C Java (<http://simplecodecjava.blogspot.com/>)

Lập trình thật đơn giản



(https://pub.accesstrade.vn/deep_link/4348611980423676010?url=http%3A%2F%2Fmytour.vn%2F)

[🏠 Bài mới \(http://simplecodecjava.blogspot.com/\)](http://simplecodecjava.blogspot.com/)

[Thuật toán \(http://simplecodecjava.blogspot.com/2015/09/thuat-toan.html\)](http://simplecodecjava.blogspot.com/2015/09/thuat-toan.html)

[Tải xuống](#) ▼

[Bài tập mẫu \(http://simplecodecjava.blogspot.com/2015/09/bai-tap-mau.html\)](http://simplecodecjava.blogspot.com/2015/09/bai-tap-mau.html)

[f \(https://www.facebook.com/simplecodecjava\)](https://www.facebook.com/simplecodecjava)

[g+ \(https://plus.google.com/115027304727901283630\)](https://plus.google.com/115027304727901283630)

[t \(https://twitter.com/SimpleCodeCJava\)](https://twitter.com/SimpleCodeCJava)

[Thuật toán] Tìm kiếm theo chiều rộng BFS.

10:36 PM (2015-08-31T22:36:00+07:00) Posted by Admin Blog (<https://plus.google.com/115027304727901283630>) No
(<https://www.blogger.com/comment.g?blogID=9046094762067454426&postID=3923482271598305952>) Comments

Để xem lý thuyết đồ thị với các định nghĩa về đường đi, chu trình, đồ thị liên thông bạn có thể xem ở [đây](http://simplecodejava.blogspot.com/2016/04/do-thi-duong-di-chi-trinh-lien-thong.html) (<http://simplecodejava.blogspot.com/2016/04/do-thi-duong-di-chi-trinh-lien-thong.html>).

Lý thuyết về thuật toán tìm kiếm theo chiều sâu bạn có thể xem ở [đây](http://simplecodejava.blogspot.com/2015/08/thuat-toan-tim-kiem-theo-chieu-sau-dfs.html).

(<http://simplecodejava.blogspot.com/2015/08/thuat-toan-tim-kiem-theo-chieu-sau-dfs.html>)

Khác với thuật toán tìm kiếm theo chiều sâu, thuật toán tìm kiếm theo chiều rộng thay thế việc sử dụng **stack** bằng hàng đợi **queue**. Trong thủ tục này, đỉnh được nạp vào hàng đợi đầu tiên là **v**, các đỉnh kề với **v** (v_1, v_2, \dots, v_k) được nạp vào queue kế tiếp. Quá trình duyệt tiếp theo được bắt đầu từ các đỉnh còn có mặt trong hàng đợi.

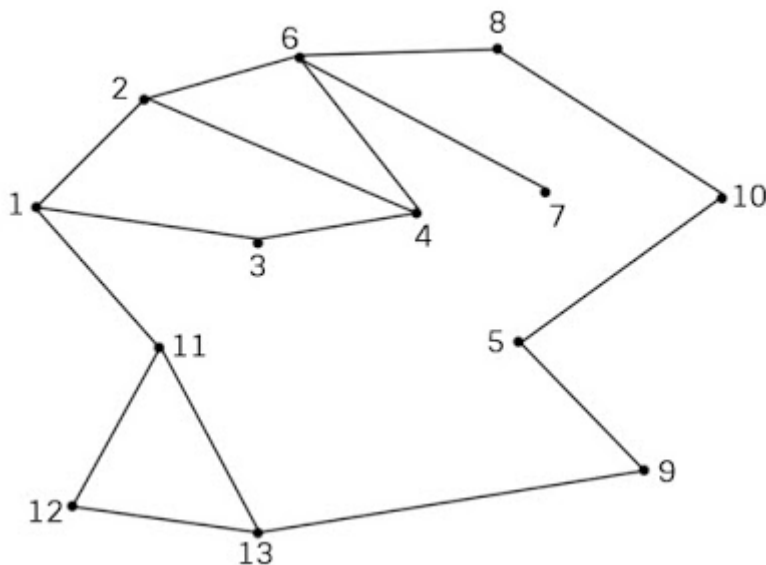
Để ghi nhận trạng thái duyệt các đỉnh của đồ thị, ta cũng vẫn sử dụng mảng **chuaxet[]** gồm **n** phần tử thiết lập giá trị ban đầu là **TRUE**. Nếu đỉnh **i** của đồ thị đã được duyệt, giá trị **chuaxet[i]** sẽ nhận giá trị **FALSE**.

Thuật toán dừng khi hàng đợi rỗng. Thủ tục **BFS** dưới đây thể hiện quá trình thực hiện của thuật toán:

```
1 void BFS(int u){
2     queue =  $\phi$ ;
3     u <= queue; /*nạp u vào hàng đợi*/
4     chuaxet[u] = false; /* đổi trạng thái của u*/
5     while (queue  $\neq \phi$ ) { /* duyệt tới khi nào hàng đợi rỗng*/
6         queue<=p; /*lấy p ra từ khỏi hàng đợi*/
7         Thăm_Đỉnh(p); /* duyệt xong đỉnh p*/
8         for (v  $\in$  ke(p) ) { /* đưa các đỉnh v kề với p nhưng chưa được xét vào hàng đợi*/
9             if (chuaxet[v] ) {
10                 v<= queue; /*đưa v vào hàng đợi*/
11                 chuaxet[v] = false; /* đổi trạng thái của v*/
12             }
13         }
14     } /* end while*/
15 } /* end BFS*/
```

Thủ tục **BFS** sẽ thăm tất cả các đỉnh cùng thành phần liên thông với **u**. Để thăm tất cả các đỉnh của đồ thị, chúng ta chỉ cần thực hiện đoạn chương trình dưới đây:

```
1 for (u=1; u<=n; u++)
2     chuaxet[u] = TRUE;
3 for (u $\in$ V )
4     if (chuaxet[u] )
5         BFS(u);
```



([http://2.bp.blogspot.com/-](http://2.bp.blogspot.com/-xE5W5vVNq8g/VeurPsmbFI/AAAAAAAAADM/bUr8628HFUY/s1600/DFS.jpg)

[xE5W5vVNq8g/VeurPsmbFI/AAAAAAAAADM/bUr8628HFUY/s1600/DFS.jpg](http://2.bp.blogspot.com/-xE5W5vVNq8g/VeurPsmbFI/AAAAAAAAADM/bUr8628HFUY/s1600/DFS.jpg))

Đồ thị - Tìm kiếm theo chiều rộng BFS

| STT | Các đỉnh đã duyệt | Các đỉnh trong hàng đợi | Các đỉnh còn lại |
|-----|-------------------------------|-------------------------|-------------------------------|
| 1 | \emptyset | \emptyset | 1,2,3,4,5,6,7,8,9,10,11,12,13 |
| 2 | 1 | 2,3,11 | 4,5,6,7,8,9,10,12,13 |
| 3 | 1,2 | 3,11,4,6 | 5,7,8,9,10,12,13 |
| 4 | 1,2,3 | 11,4,6 | 5,7,8,9,10,12,13 |
| 5 | 1,2,3,11 | 4,6,12,13 | 5,7,8,9,10 |
| 6 | 1,2,3,11,4 | 6,12,13 | 5,7,8,9,10 |
| 7 | 1,2,3,11,4,6 | 12,13,7,8 | 5,9,10 |
| 8 | 1,2,3,11,4,6,12 | 13,7,8 | 5,9,10 |
| 9 | 1,2,3,11,4,6,12,13 | 7,8,9 | 5,10 |
| 10 | 1,2,3,11,4,6,12,13,7 | 8,9 | 5,10 |
| 11 | 1,2,3,11,4,6,12,13,7,8 | 9,10 | 5 |
| 12 | 1,2,3,11,4,6,12,13,7,8,9 | 10,5 | \emptyset |
| 13 | 1,2,3,11,4,6,12,13,7,8,9,10 | 5 | \emptyset |
| 14 | 1,2,3,11,4,6,12,13,7,8,9,10,5 | \emptyset | \emptyset |

Kết quả duyệt: 1,2,3,11,4,6,12,13,7, 8, 9,10, 5.

Mã trận liên kề được tải về tại [đây](https://www.dropbox.com/s/8two2mhp1ul5hn/BFS.IN?dl=0). (<https://www.dropbox.com/s/8two2mhp1ul5hn/BFS.IN?dl=0>)

Chương trình cài đặt bằng C/C++

```

1  #include<iostream>
2  #include<conio.h>
3  using namespace std;
4  #define MAX 100
5  #define TRUE 1
6  #define FALSE 0
7  int G[MAX][MAX], n, chuaxet[MAX], QUEUE[MAX];
8  void Init(){
9      freopen("BFS.IN", "r", stdin);
10     cin>>n;
11     cout<<"So dinh cua do thi n = "<<n<<endl;
12     //nhap ma tran lien ke.
13     for(int i=1; i<=n;i++){
14         for(int j=1; j<=n;j++){
15             cin>>G[i][j];
16         }
17     }
18     for(int i=1; i<=n;i++){
19         chuaxet[i]=TRUE;
20     }
21 }
22 /* Breadth First Search */
23 void BFS(int i){
24     int u,dauQ, cuoiQ;
25     dauQ=1; cuoiQ=1;QUEUE[cuoiQ]=i;chuaxet[i]=FALSE;
26     /* thiết lập hàng đợi với đỉnh đầu là i*/
27     while(dauQ<=cuoiQ){
28         u=QUEUE[dauQ];// lấy đỉnh u ra khỏi queue.
29         cout<<"duyet dinh: "<<u<<endl;
30         dauQ=dauQ+1; /* duyệt đỉnh đầu hàng đợi*/
31         for(int j=1; j<=n;j++){
32             if(G[u][j]==1 && chuaxet[j] ){
33                 cuoiQ=cuoiQ+1;
34                 QUEUE[cuoiQ]=j;
35                 chuaxet[j]=FALSE;
36             }
37         }
38     }
39 }
40 void main(void){
41     Init();
42     for(int i=1; i<=n; i++)
43         if (chuaxet[i])
44             BFS(i);
45     _getch();
46 }

```

Chương trình cài đặt bằng Java

```

1  package simplecodejava.blogspot.com;
2
3  import java.io.BufferedReader;
4  import java.io.File;
5  import java.io.FileNotFoundException;
6  import java.io.FileReader;
7  import java.io.IOException;
8
9  public class BFSAlgorithm {
10     static final int MAX = 100;
11     int G[][] = new int[MAX][MAX];/* ma trận kề*/
12     boolean chuaxet[] = new boolean[MAX];/*mảng đánh dấu đỉnh đã được thăm.*/
13     int QUEUE[] = new int[MAX]; /*hàng đợi*/
14     int n;
15
16     void init() {
17         File file = new File(getClass().getResource("BFS.IN").getPath());
18         BufferedReader reader = null;

```

```

19     try {
20         reader = new BufferedReader(new FileReader(file));
21         n = Integer.parseInt(reader.readLine());
22         for (int i = 1; i <= n; i++) {
23             String[] aLineOfMatrix = reader.readLine().split("\\s+");
24             for (int j = 1; j <= n; j++) {
25                 G[i][j] = Integer.parseInt(aLineOfMatrix[j - 1]); /*index của J bắt đầu từ 0*/
26             }
27         }
28     } catch (FileNotFoundException e) {
29         e.printStackTrace();
30     } catch (IOException e) {
31         e.printStackTrace();
32     } finally {
33         if (reader != null)
34             try {
35                 reader.close();
36             } catch (IOException e) {
37                 e.printStackTrace();
38             }
39     }
40 }

41
42 void BFS(int v) {
43     int u, dauQ, cuoiQ, j;
44     dauQ = 1;
45     cuoiQ = 1;
46     QUEUE[cuoiQ] = v;
47     chuaxet[v] = false;
48     /* thiết lập hàng đợi với đỉnh đầu là i */
49     while (dauQ <= cuoiQ) {
50         u = QUEUE[dauQ];
51         System.out.print(u + " ");
52         dauQ = dauQ + 1; /* duyệt đỉnh đầu hàng đợi */
53         for (j = 1; j <= n; j++) {
54             if (G[u][j] == 1 && chuaxet[j]) {
55                 cuoiQ = cuoiQ + 1;
56                 QUEUE[cuoiQ] = j;
57                 chuaxet[j] = false;
58             }
59         }
60     }
61 }

62
63 public BFSAlgorithm() {
64     init();
65     for (int i = 1; i <= n; i++) {
66         chuaxet[i] = true;
67     }
68     System.out.print("\n");
69     for (int i = 1; i <= n; i++)
70         if (chuaxet[i]) {
71             BFS(i);
72         }
73 }

74
75 public static void main(String[] args) {
76     new BFSAlgorithm();
77 }
78 }

```

Share this

✚ Google (<https://plus.google.com/share?url=http://simplecodejava.blogspot.com/2015/08/thuat-toan-tim-kiem-theo-chieu-rong-bfs.html>)

Facebook (<https://www.facebook.com/sharer/sharer.php?u=http://simplecodejava.blogspot.com/2015/08/thuat-toan-tim-kiem-theo-chieu-rong-bfs.html>)

Twitter (<https://twitter.com/intent/tweet?text=%5BThu%E1%BA%ADt%20to%C3%A1n%5D%20T%C3%ACm%20ki%E1%BA%BFm%20theo%20chi%E1%BB%81u%20r%E1%BB%99ng%20BFS.%20-%20Simple%20Code%20C%20Java&url=http://simplecodejava.blogspot.com/2015/08/thuat-toan-tim-kiem-theo-chieu-rong-bfs.html>)

+ More



Chào mừng bạn đến với SimpleCodeJava Blog - Mục đích của chúng tôi khi thành lập blog này là muốn chia sẻ những kiến thức và kinh nghiệm lập trình mà chúng tôi đã học được với mong muốn giúp đỡ mọi người, giúp bạn rút ngắn được thời gian tìm hiểu cũng như việc giải quyết những vấn đề trong lập trình C và Java.

Bài liên quan

✚ [\[Thuật toán\] Tìm kiếm theo chiều rộng BFS. \(http://simplecodejava.blogspot.com/2015/08/thuat-toan-tim-kiem-theo-chieu-rong-bfs.html\)](http://simplecodejava.blogspot.com/2015/08/thuat-toan-tim-kiem-theo-chieu-rong-bfs.html)

Tag : [BFS \(http://simplecodejava.blogspot.com/search/label/BFS\)](http://simplecodejava.blogspot.com/search/label/BFS) , [đồ thị \(http://simplecodejava.blogspot.com/search/label/%C4%91%E1%BB%93%20th%E1%BB%8B\)](http://simplecodejava.blogspot.com/search/label/%C4%91%E1%BB%93%20th%E1%BB%8B) , [Tìm kiếm theo chiều rộng \(http://simplecodejava.blogspot.com/search/label/T%C3%ACm%20ki%E1%BA%BFm%20theo%20chi%E1%BB%81u%20r%E1%BB%99ng\)](http://simplecodejava.blogspot.com/search/label/T%C3%ACm%20ki%E1%BA%BFm%20theo%20chi%E1%BB%81u%20r%E1%BB%99ng)

← Newer Post (<http://simplecodejava.blogspot.com/2015/09/thuat-toan-duyet-cac-thanh-phan-lien.html>) [Home](#) (<http://simplecodejava.blogspot.com/2015/08/thuat-toan-tim-kiem-theo-chieu-sau-dfs.html>)
Older Post → (<http://simplecodejava.blogspot.com/>)

0 bình luận

Sắp xếp theo

Cũ nhất



Thêm bình luận...

Plugin bình luận của Facebook

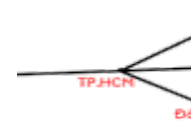
0 Comment to "[Thuật toán] Tìm kiếm theo chiều rộng BFS."

Subscribe to: [Post Comments \(Atom\)](#)

(<http://simplecodejava.blogspot.com/feeds/3923482271598305952/comments/default>)

RECENT ★ WEEKLY

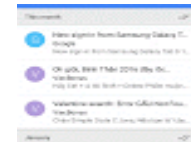
COMMENT



[Đồ thị] Lý thuyết đồ thị - Đường đi, Chu trình, Đồ thị liên thông

(<http://simplecodejava.blogspot.com/2016/04/do-thi-duong-di-chi-trinh-lien-thong.html>)

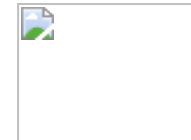
Đồ thị: là một cấu trúc dữ liệu rời rạc bao...



[Android] Swipe RecyclerView cho Android

(<http://simplecodejava.blogspot.com/2016/02/android-swipe-recyclerview-cho-android.html>)

Sau một thời gian sử dụng sử dụng Gmail và...



[Android] Thư viện TimePicker & DatePicker đẹp cho android.

(<http://simplecodejava.blogspot.com/2016/02/android-thu-vien-timepicker-datepicker.html>)

Qua một thời gian sử dụng Google calendar mình...



[Java] Copy dữ liệu sang file khác trong Java

(<http://simplecodejava.blogspot.com/2016/01/java-copy-du-lieu-sang-file-khac-trong-java.html>)

Bài viết sẽ trình bày cách copy dữ liệu từ một...



[Java] Lấy IP trong Java

(<http://simplecodejava.blogspot.com/2016/01/java-lay-ip-trong-java.html>)

Bài viết sẽ hướng dẫn bạn lấy địa chỉ IP từ...



[Java] Chuyển từ hệ thập phân sang hệ nhị phân.
(<http://simplecodej.ava.blogspot.com/2016/01/java-chuyen-tu-he-thap-phan-sang-he-nhi-phan.html>)

ava.blogspot.com/2016/01/java-chuyen-tu-he-thap-phan-sang-he-nhi.html)

Có 3 cách để chuyển một số từ hệ thập phân...



[Java] Sự khác nhau giữa HashMap và Hashtable
(<http://simplecodej.ava.blogspot.com/2016/01/java-su-khac-nhau-giua-hashmap-va-hashtable.html>)

ava.blogspot.com/2016/01/java-su-khac-nhau-giua-hashmap-va.html)

Sự khác nhau giữa HashMap và Hashtable ? Đây...



[Java] Tạo file trong Java
(<http://simplecodej.ava.blogspot.com/2015/12/java-tao-file-trong-java.html>)

15/12/java-tao-file-trong-java.html)
Bài viết này sẽ trình bày cách tạo một file trong...

FOLLOW US ON FACEBOOK



Simple Code C Java
394 likes

Like Page

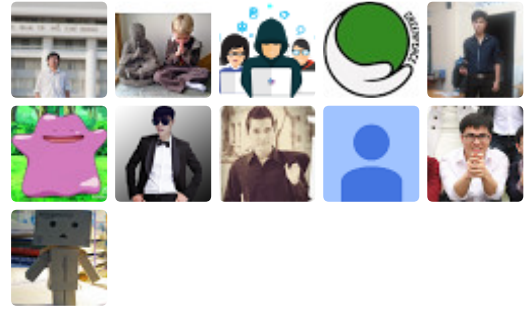
Be the first of your friends to like this



GOOGLE+ FOLLOWERS

Admin Blog

Add to circles



13 have me in circles

[View all](#)

FOLLOW BY EMAIL

Email address...

Submit

Copyright © 2018 [Simple Code C Java \(http://simplecodejava.blogspot.com/\)](http://simplecodejava.blogspot.com/) Distributed By [Blogger Templates \(http://www.protemplateslab.com/\)](http://www.protemplateslab.com/) | Theme By [Arlina Design \(http://arlinadesign.blogspot.com/\)](http://arlinadesign.blogspot.com/)