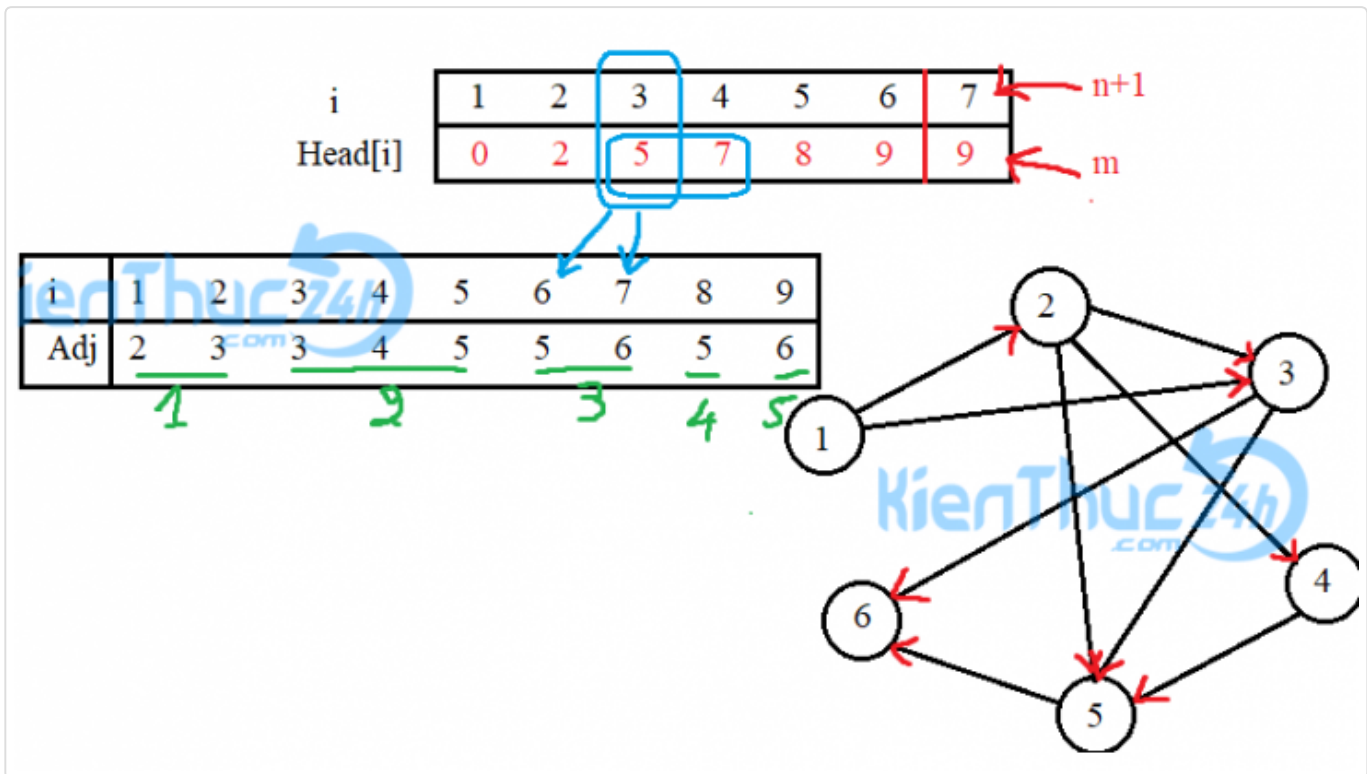


[Trang chủ](#)
[Series](#)
[Thuật toán](#)
[Ngành CNTT](#)
[Ngôn ngữ](#)
[Máy chủ - VPS](#)
[Văn học](#)

[Webmaster](#)
[Themis web](#)
[Tools](#)
[Contact](#)



Bài 3: Danh sách kề C++ Lý thuyết đồ thị

📅 7 Tháng Tám, 2017 (<https://kienthuc24h.com/danh-sach-ke-c-ly-thuyet-do-thi/>)
 👤 Đặng Minh Tiến (<https://kienthuc24h.com/author/miticc06/>)

f (<https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fkienthuc24h.com%2Fdanh-sach-ke-c-ly-thuyet-do-thi/>)

t (<https://twitter.com/intent/tweet?text=&url=https%3A%2F%2Fkienthuc24h.com%2Fdanh-sach-ke-c-ly-thuyet-do-thi/>)

G+ (<https://plus.google.com/share?url=https%3A%2F%2Fkienthuc24h.com%2Fdanh-sach-ke-c-ly-thuyet-do-thi/>)

Bài viết này là phần 3 trong 7 bài của Series Lý thuyết đồ thị căn bản (<https://kienthuc24h.com/series/ly-thuyet-do-thi-can-ban/>)

Lý thuyết đồ thị căn bản (<https://kienthuc24h.com/series/ly-thuyet-do-thi-can-ban/>)

Bài 1: Ma trận kề C++/Pascal Lý thuyết đồ thị (<https://kienthuc24h.com/ma-tran-ke-c-pascal-ly-thuyet-thi/>)

Bài 2: Danh sách cạnh C++ Lý thuyết đồ thị (<https://kienthuc24h.com/danh-sach-can-c-ly-thuyet-do-thi/>)

Bài 3: Danh sách kề C++ Lý thuyết đồ thị

Bài 4: Thuật toán tìm kiếm theo chiều sâu DFS pascal c++ (<https://kienthuc24h.com/thuat-toan-tim-kiem-theo-chieu-sau-dfs/>)

Bài 5: Thuật toán tìm kiếm theo chiều rộng BFS pascal c++ (<https://kienthuc24h.com/thuat-toan-tim-kiem-theo-chieu-rong-bfs/>)

Để khắc phục yếu điểm của danh sách cạnh về việc tìm đỉnh kề, mà vẫn đảm bảo tổ chức dữ liệu tối ưu nhất phục vụ duyệt tìm trong đồ thị mà Danh sách kề được ra đời.

Nội dung bài viết [hide]

1. Ý tưởng danh sách kề
 - a. Ý tưởng
 - a.1 Tổ chức bằng mảng
 - a.2 Tổ chức bằng danh sách liên kết
 - b. Ví dụ minh họa
 - c. Độ phức tạp
2. Cài đặt danh sách kề
 - a. Tổ chức bằng mảng bằng pascal / C++
 - b. Tổ chức danh sách liên kết
3. Ưu điểm và hạn chế của danh sách kề
4. Bài tập ứng dụng

1. Ý tưởng danh sách kề

a. Ý tưởng

a.1 Tổ chức bằng mảng

Tổ chức dữ liệu kiểu danh sách kề là bạn sẽ lưu các đỉnh kề của một đỉnh vào các đoạn trong mảng để lưu trữ. Nếu có N đỉnh thì sẽ có N đoạn để lưu. Và chúng ta cần lưu lại chỉ số để quản lý phạm vi các đỉnh kề của đỉnh mà đoạn đó quản lý.

Các bạn xem hình ảnh bên dưới để hiểu rõ hơn ý tưởng này.

Gọi $\text{Head}[i]$ là chỉ số kết thúc của đoạn quản lý đỉnh kề của $i-1$. Hay $\text{Head}[i]+1$ là chỉ số bắt đầu của đoạn quản lý đỉnh kề của i .

Như vậy, chỉ số của đoạn chứa các đỉnh kề của i là từ **$\text{Head}[i]+1$ đến $\text{Head}[i+1]$**

a.2 Tổ chức bằng danh sách liên kết

Việc tổ chức bằng danh sách liên kết, chúng ta sẽ có một mảng n ô, Mỗi ô chứa một pointer Node, pointer này gồm 2 thông tin là chỉ số đỉnh kề và pointer next tiếp theo. Tham khảo hình bên dưới (mục cài đặt bằng danh sách liên kết) để hiểu rõ hơn.

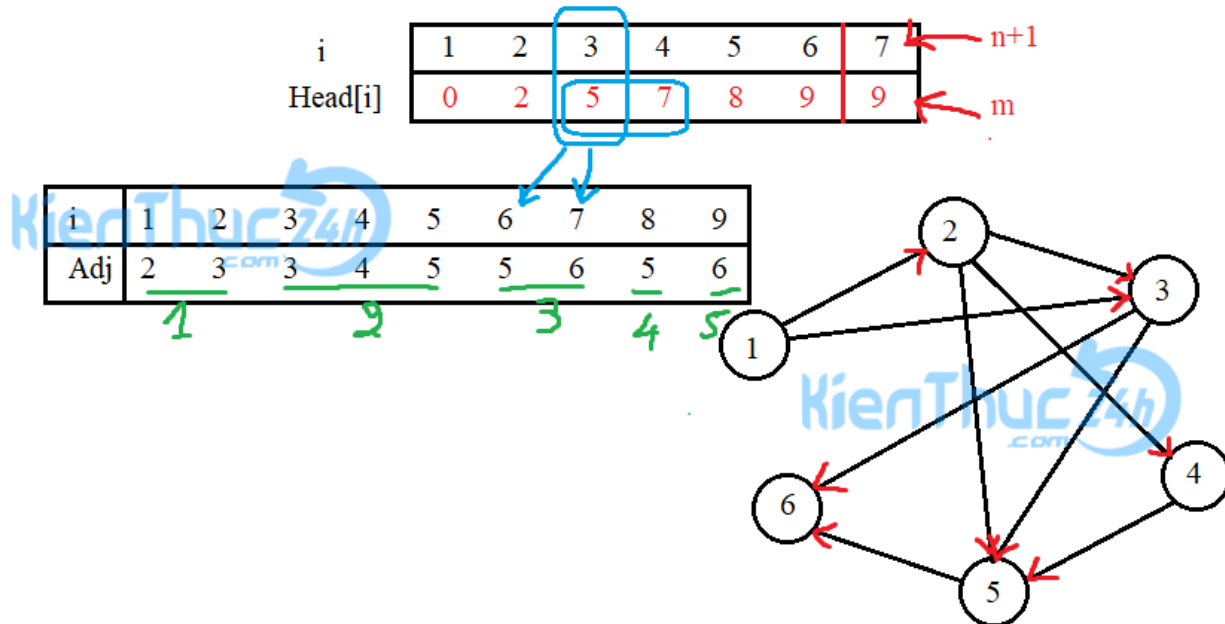
b. Ví dụ minh họa

Cho đồ thị đơn có hướng, có N đỉnh M cạnh. Dữ liệu các cạnh như sau:

```

1 1 2
2 1 3
3 2 4
4 2 3
5 2 5
6 3 5
7 3 6
8 4 5
9 5 6

```



Ví dụ minh họa tổ chức danh sách kề bằng mảng một chiều

Trước tiên bạn chỉ cần hiểu **các đỉnh kề của đỉnh u** là **Adj[v]** với **v=Head[u]+1.....Head[u+1]**

Như hình minh họa bên trên ở đỉnh số 3, mình có tô màu xanh ví dụ. Dựa vào việc xác định đỉnh kề bằng Head[] thì mình có được các đỉnh kề của 3 là Adj[Head[3]+1 đến Head[3+1]]

Nếu viết bằng c++ thì mình lấy các đỉnh kề của đỉnh u bằng cách sau:

```

1 // Minh họa in các đỉnh kề của u
2 void indinhke(int u)
3 {
4     for (int v=Head[u]+1; v<=Head[u+1]; v++)
5         cout << Adj[v]<<endl;
6 }

```

c. Độ phức tạp

Độ phức tạp dữ liệu và thời gian là $O(m)$ với m là số cạnh của đồ thị.

^

2. Cài đặt danh sách kề

a. Tổ chức bằng mảng bằng pascal / C++

Như những bài viết trước về tổ chức dữ liệu cho lý thuyết đồ thị, mình sẽ hướng dẫn các bạn đọc dữ liệu. Dòng đầu sẽ gồm 2 số N, và m.

Mô tả từng bước:

- Ban đầu Head[i]=0
- Đọc từng cạnh (u,v) và tăng Head[u]=Head[u]+1.
- Cộng dồn Head[i]=Head[i-1]+Head[i]. i=2->n+1;
- Lưu ý thêm Head[n+1] luôn m nếu đó là đồ thị có hướng.
- Duyệt qua các cạnh, Adj[Head[u]] = v; Head[u]=Head[u]-1;

Code tổ chức dữ liệu đồ thị bằng danh sách kề trong c++

```

1  #define NMAX 100
2  #define MMAX 1000
3
4  //Khai báo dữ liệu
5  int x[MMAX], y[MMAX], Head[NMAX+2], Adj[MMAX+1];
6  int n, m, u, v;
7
8  // Đọc dữ liệu
9  cin >> n >> m;
10
11 // Gán Head[i]=0
12 for (int i=0; i<=n+1; i++)
13     Head[i]=0;
14
15 for (int i=1; i<=m; i++)
16 {
17     cin >> x[i] >> y[i];
18     Head[x[i]]++;
19 }
20
21 for (int i=2; i<=n+1; i++)
22     Head[i]=Head[i-1]+Head[i];
23
24 for (int i=1; i<=m; i++)
25 {
26     Adj[Head[x[i]]]=y[i];
27     Head[x[i]]--;
28 }

```

Code tổ chức dữ liệu đồ thị bằng danh sách kề trong Pascal

```

1  const nmax=100;
2      mmax=1000;
3  type data=longint;
4  // khai bao du lieu
5  var
6      adj:array[0..mmax] of data;
7      head,u,v:array[0..mmax+1] of data;
8
9  //doc du lieu
10 procedure docfile;
11 var
12     i,j:data;
13 begin
14     read(n,m);
15     fillchar(head,sizeof(head),0);
16     for i:=1 to m do
17     begin
18         read(f,u[i],v[i]);
19         inc(head[u[i]]);
20     end;
21     for i:=2 to n+1 do
22         head[i]:=head[i-1]+head[i];
23
24     for i:=1 to m do
25     begin
26         adj[head[u[i]]]:=v[i];
27         dec(head[u[i]]);
28     end;
29 end;

```

Code bên trên là lưu trữ cạnh có hướng, nếu bạn muốn cạnh vô hướng bạn có thể xử lý thêm:

```

1 // Lúc nhập dữ liệu
2 Head[x[i]]++;
3 Head[y[i]]++;

```

Sau khi cộng dồn head, và tất nhiên MMAX phải tăng gấp đôi.

```

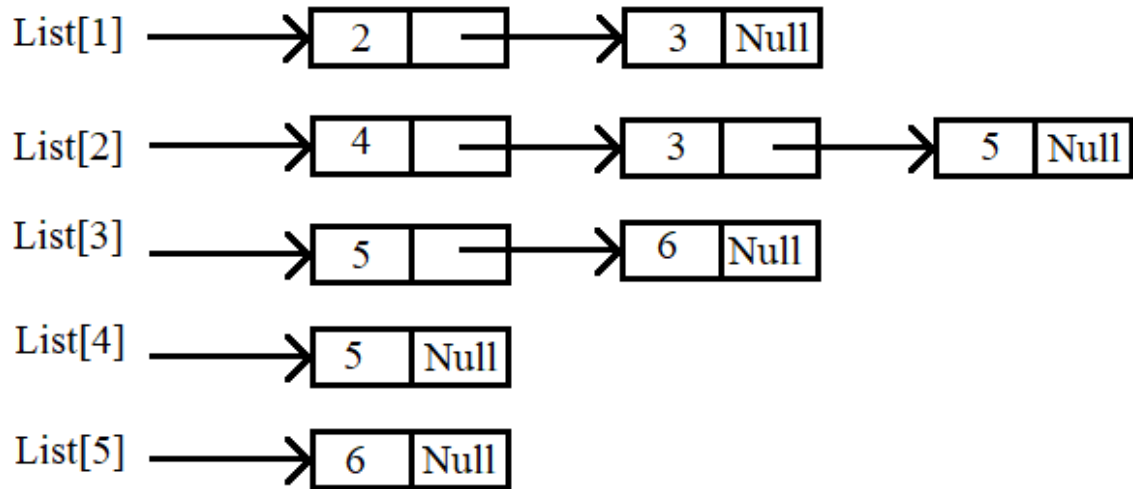
1 for (int i=1; i<=m; i++)
2 {
3     Adj[Head[x[i]]]=y[i];
4     Head[x[i]]--;
5
6     Adj[Head[y[i]]]=x[i];
7     Head[y[i]]--;
8 }

```

Nếu bạn muốn lưu thêm trọng số của cạnh, có thể tạo thêm một mảng có vai trò giống như adj, như lúc này chúng ta sẽ lưu trọng số của chúng.

b. Tổ chức danh sách liên kết





Tổ chức lưu trữ đồ thị bằng danh sách liên kết

```

1  struct Node
2  {
3      int info;
4      Node* pNext;
5      Node()
6      {
7          pNext=NULL;
8      }
9  }
10
11 Node* List[NMAX];
12 int u,v,m,n;
13
14 // Nhập số đỉnh và số cạnh
15 cin >> n >> m;
16 for (int i=1; i<=m; i++)
17 {
18     cin >> u >> v;
19     Node *p = new Node();
20     p->info = v;
21
22     if (List[u]==NULL)
23         List[u] = p;
24     else
25     {
26         p->pNext = List[u];
27         List[u] = p;
28     }
29 }
  
```

Duyệt đỉnh kề của u trong danh sách liên kết như thế nào ?

```

1 // Minh họa in các đỉnh kề của u
2 void indinhke(int u)
3 {
4     Node *p = List[u];
5     while (p!=NULL)
6     {
7         cout << p->info << endl;
8         p=p->pNext;
9     }
10 }

```

3. Ưu điểm và hạn chế của danh sách kề

- Ưu điểm của danh sách kề là chi phí duyệt và lưu trữ khá tối ưu. Đặc biệt là danh sách kề trong mảng.
- Cài đặt bài toán bằng danh sách kề tương đối dài hơn so với ma trận kề và danh sách cạnh.
- Đối với từng bài toán cụ thể, tùy vào dữ liệu bài toán cho, các bạn hãy lựa chọn cách tổ chức dữ liệu phù hợp nhất, không nhất thiết lúc nào cũng tổ chức danh sách kề.

4. Bài tập ứng dụng

Bài tập về danh sách kề nhìn chung các bạn có thể lấy các bài tập mình share ở bài về DFS (<https://kienthuc24h.com/thuat-toan-tim-kiem-theo-chieu-sau-dfs/>), BFS (<https://kienthuc24h.com/thuat-toan-tim-kiem-theo-chieu-rong-bfs/>) về bản chất nó là như nhau.

Ngoài ra có thể tham khảo danh sách kề khi kết hợp với Prim

heap: <https://kienthuc24h.com/cay-khung-nho-nhat-qbmst-spoj-kruskal-prim-heap/>
(<https://kienthuc24h.com/cay-khung-nho-nhat-qbmst-spoj-kruskal-prim-heap/>)

Bài viết liên quan

- Bài 2: Danh sách cạnh C++ Lý thuyết đồ thị (<https://kienthuc24h.com/danh-sach-can-h-c-ly-thuyet-do-thi/>)
- Bài 1: Ma trận kề C++/Pascal Lý thuyết đồ thị (<https://kienthuc24h.com/ma-tran-ke-cpascal-ly-thuyet-thi/>)
- Bài 6: Thuật toán loang trên ma trận (<https://kienthuc24h.com/thuat-toan-loang-tren-ma-tran/>)
- [BFS] – SPOJ PPATH (<https://kienthuc24h.com/bfs-spoj-ppath/>)
- Bài 4: Thuật toán tìm kiếm theo chiều sâu DFS pascal c++ (<https://kienthuc24h.com/thuat-toan-tim-kiem-theo-chieu-sau-dfs/>)
- Bài 5: Thuật toán tìm kiếm theo chiều rộng BFS pascal c++ (<https://kienthuc24h.com/thuat-toan-tim-kiem-theo-chieu-rong-bfs/>)
- [Upcoder BFS] r2.b3.Hereditament – Hereditament (<https://kienthuc24h.com/upcoder-bfs-r2-b3-hereditament-hereditament/>)

- Cây khung nhỏ nhất QBMST spoj: Kruskal, Prim heap (<https://kienthuc24h.com/cay-khung-nho-nhat-qbmst-spoj-kruskal-prim-heap/>)
- VDANGER SPOJ- Nguy hiểm rõ ràng trước mắt (<https://kienthuc24h.com/vdanger-spoj-nguy-hiem-ro-rang-truoc-mat/>)
- CRITICAL SPOJ- Thành phố trọng yếu (<https://kienthuc24h.com/critical-thanh-pho-trong-yeu/>)

📁 Đồ thị (<https://kienthuc24h.com/category/thuat-toan/do-thi/>) 🗨️ Tổ chức dữ liệu trong lý thuyết đồ thị (<https://kienthuc24h.com/tag/to-chuc-du-lieu-trong-ly-thuyet-do-thi/>). 🔗 permalink (<https://kienthuc24h.com/danh-sach-ke-c-ly-thuyet-do-thi/>).

◀ Viết mã nguồn tự động thông báo điểm UIT (Phần 2) (<https://kienthuc24h.com/viet-ma-nguon-tu-dong-thong-bao-diem-uit-phan-2/>)

Cách đọc ghi file trong c++ ➤ (<https://kienthuc24h.com/cach-doc-ghi-file-trong-c/>)

2 thoughts on “Bài 3: Danh sách kề C++ Lý thuyết đồ thị”

1.



quang says:

7 Tháng Tám, 2017 at 21:57 (<https://kienthuc24h.com/danh-sach-ke-c-ly-thuyet-do-thi/#comment-560>)

trời ơi hay quá :((hay quá hay quá :((hay kinh.

Trả lời (<https://kienthuc24h.com/danh-sach-ke-c-ly-thuyet-do-thi/?replytocom=560#respond>)

2.



kudoshinichilike says:

7 Tháng Tám, 2017 at 22:13 (<https://kienthuc24h.com/danh-sach-ke-c-ly-thuyet-do-thi/#comment-561>)

hay hay hay! <3

Trả lời (<https://kienthuc24h.com/danh-sach-ke-c-ly-thuyet-do-thi/?replytocom=561#respond>)

Trả lời

^

Email của bạn sẽ không được hiển thị công khai. Các trường bắt buộc được đánh dấu *

Bình luận

Tên *

Email *

Trang web

Phản hồi

Email Subscribe

Hãy Subscribe và bình luận tích cực để chúng tôi có thêm động lực phát triển KienThuc24h.com

Email *

Subscribe

GẦN ĐÂY

PHỔ BIẾN



[CSDL] BT Thực hành 1 truy vấn SQL – Phần 2 (<https://kienthuc24h.com/cSDL-bt-thuc-hanh-1-truy-van-sql-phan-2/>)

Th12 16, 2017



Share post Facebook để mở khóa nội dung bằng cách sử dụng jQuery & Facebook API (<https://kienthuc24h.com/share-post-facebook-de-mo-khoa-noi-dung-bang-cach-su-dung-jquery-facebook-api/>)

Th12 2, 2017

[Stack] PostFix to InFix (<https://kienthuc24h.com/stack-postfix-infix/>)



Th11 30, 2017

[CSDL] BT Thực hành 1 truy vấn SQL – Phần 1 (<https://kienthuc24h.com/csdl-bt-thuc-hanh-1-truy-van-sql/>)

Th11 26, 2017

[Stack]- SPOJ ONP – Transfer the expression – Infix to Postfix (<https://kienthuc24h.com/stack-spoj-onp-transfer-expression-infix-postfix/>)

Th11 21, 2017

Chuyên mục

📁 Chưa được phân loại (<https://kienthuc24h.com/category/khong-phan-loai/>) (15)

📁 Máy chủ – VPS (<https://kienthuc24h.com/category/may-chu-vps/>) (6)

📁 Ngành CNTT (<https://kienthuc24h.com/category/nganh-cntt/>) (43)

📁 Cơ sở dữ liệu (<https://kienthuc24h.com/category/nganh-cntt/co-so-du-lieu/>) (3)

📁 CTDL & Giải thuật (<https://kienthuc24h.com/category/nganh-cntt/ctdl-giai-thuat/>) (2)

📁 Lập trình hướng đối tượng OOP (<https://kienthuc24h.com/category/nganh-cntt/lap-trinh-huong-doi-tuong-oop/>) (15)

📁 Nhập môn lập trình (<https://kienthuc24h.com/category/nganh-cntt/nhap-mon-lap-trinh/>) (21)

📁 Ngôn ngữ (<https://kienthuc24h.com/category/ngon-ngu/>) (22)

📁 C++ (<https://kienthuc24h.com/category/ngon-ngu/code-c/>) (16)

📁 Pascal (<https://kienthuc24h.com/category/ngon-ngu/pascal/>) (6)

📁 Sự kiện (<https://kienthuc24h.com/category/su-kien/>) (2)

📁 Thuật toán (<https://kienthuc24h.com/category/thuat-toan/>) (307)

📁 Bitmask (<https://kienthuc24h.com/category/thuat-toan/bitmask/>) (3)

📁 Cấu trúc dữ liệu (<https://kienthuc24h.com/category/thuat-toan/cau-truc-du-lieu/>) (26)

📁 Binary Indexed Tree (<https://kienthuc24h.com/category/thuat-toan/cau-truc-du-lieu/binary-indexed-tree/>) (2)

📁 Hash (<https://kienthuc24h.com/category/thuat-toan/cau-truc-du-lieu/hash/>) (2)

📁 Interval Tree (<https://kienthuc24h.com/category/thuat-toan/cau-truc-du-lieu/interval-tree/>) (6)

📁 Queue (<https://kienthuc24h.com/category/thuat-toan/cau-truc-du-lieu/queue/>) (1)

📁 Stack (<https://kienthuc24h.com/category/thuat-toan/cau-truc-du-lieu/stack/>) (13)

📁 Duyệt (<https://kienthuc24h.com/category/thuat-toan/duyet/>) (86)

📁 [Đệ quy & quay lui](https://kienthuc24h.com/category/thuat-toan/de-quy-quay-lui/) (<https://kienthuc24h.com/category/thuat-toan/de-quy-quay-lui/>) (18)

📁 [Đồ thị](https://kienthuc24h.com/category/thuat-toan/do-thi/) (<https://kienthuc24h.com/category/thuat-toan/do-thi/>) (51)

📁 [BFS – DFS](https://kienthuc24h.com/category/thuat-toan/do-thi/bfs-dfs/) (<https://kienthuc24h.com/category/thuat-toan/do-thi/bfs-dfs/>) (25)

📁 [Cặp ghép](https://kienthuc24h.com/category/thuat-toan/do-thi/cap-ghep/) (<https://kienthuc24h.com/category/thuat-toan/do-thi/cap-ghep/>) (4)

📁 [Cầu](https://kienthuc24h.com/category/thuat-toan/do-thi/cau/) (<https://kienthuc24h.com/category/thuat-toan/do-thi/cau/>) (1)

📁 [Cây khung](https://kienthuc24h.com/category/thuat-toan/do-thi/cay-khung/) (<https://kienthuc24h.com/category/thuat-toan/do-thi/cay-khung/>) (7)

📁 [Floyd – Dijkstra](https://kienthuc24h.com/category/thuat-toan/do-thi/floyd-dijkstra/) (<https://kienthuc24h.com/category/thuat-toan/do-thi/floyd-dijkstra/>) (3)

📁 [Khớp](https://kienthuc24h.com/category/thuat-toan/do-thi/khop/) (<https://kienthuc24h.com/category/thuat-toan/do-thi/khop/>) (3)

📁 [Liên thông](https://kienthuc24h.com/category/thuat-toan/do-thi/lien-thong/) (<https://kienthuc24h.com/category/thuat-toan/do-thi/lien-thong/>) (11)

📁 [Luồng](https://kienthuc24h.com/category/thuat-toan/do-thi/luong/) (<https://kienthuc24h.com/category/thuat-toan/do-thi/luong/>) (1)

📁 [Heap](https://kienthuc24h.com/category/thuat-toan/heap/) (<https://kienthuc24h.com/category/thuat-toan/heap/>) (2)

📁 [Hình học](https://kienthuc24h.com/category/thuat-toan/hinh-hoc/) (<https://kienthuc24h.com/category/thuat-toan/hinh-hoc/>) (4)

📁 [Nhân ma trận](https://kienthuc24h.com/category/thuat-toan/nhan-ma-tran/) (<https://kienthuc24h.com/category/thuat-toan/nhan-ma-tran/>) (2)

📁 [Quy hoạch động](https://kienthuc24h.com/category/thuat-toan/quy-hoach-dong/) (<https://kienthuc24h.com/category/thuat-toan/quy-hoach-dong/>) (36)

📁 [Sắp xếp](https://kienthuc24h.com/category/thuat-toan/sap-xep/) (<https://kienthuc24h.com/category/thuat-toan/sap-xep/>) (26)

📁 [Số học](https://kienthuc24h.com/category/thuat-toan/so-hoc/) (<https://kienthuc24h.com/category/thuat-toan/so-hoc/>) (25)

📁 [Tìm kiếm nhị phân](https://kienthuc24h.com/category/thuat-toan/tim-kiem-nhi-phan/) (<https://kienthuc24h.com/category/thuat-toan/tim-kiem-nhi-phan/>) (14)

📁 [Xử lý số lớn](https://kienthuc24h.com/category/thuat-toan/xu-li-so-lon/) (<https://kienthuc24h.com/category/thuat-toan/xu-li-so-lon/>) (5)

📁 [xử lý xấu](https://kienthuc24h.com/category/thuat-toan/xu-li-xau/) (<https://kienthuc24h.com/category/thuat-toan/xu-li-xau/>) (53)

📁 [Văn học](https://kienthuc24h.com/category/van-hoc/) (<https://kienthuc24h.com/category/van-hoc/>) (20)

📁 [Webmaster](https://kienthuc24h.com/category/webmaster/) (<https://kienthuc24h.com/category/webmaster/>) (12)

📁 [Mã nguồn HTML](https://kienthuc24h.com/category/webmaster/ma-nguon-html/) (<https://kienthuc24h.com/category/webmaster/ma-nguon-html/>) (1)

📁 [Mã nguồn Javascript](https://kienthuc24h.com/category/webmaster/ma-nguon-javascript/) (<https://kienthuc24h.com/category/webmaster/ma-nguon-javascript/>) (1)

📁 [Mã nguồn PHP](https://kienthuc24h.com/category/webmaster/ma-nguon-php/) (<https://kienthuc24h.com/category/webmaster/ma-nguon-php/>) (4)

📁 [Mã nguồn wordpress](https://kienthuc24h.com/category/webmaster/ma-nguon-wordpress/) (<https://kienthuc24h.com/category/webmaster/ma-nguon-wordpress/>) (2)

Copyright by KienThuc24h.Com

[Privacy Policy](https://kienthuc24h.com/privacy-policy/) (<https://kienthuc24h.com/privacy-policy/>) - [Terms and Conditions](https://kienthuc24h.com/terms-and-conditions/) (<https://kienthuc24h.com/terms-and-conditions/>)