**ÔN TẬP**

**CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

HK183

**CHƯƠNG 1: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**1)** Tìm một số ví dụ minh hoạ mối quan hệ giữa cấu trúc dữ liệu và giải thuật. Giải thích minh họa nếu cần.

**2)** Cho biết một số kiểu dữ liệu được định nghĩa sẵn trong một ngôn ngữ lập trình các bạn thường sử dụng. Cho biết một số kiểu dữ liệu tiền định này có đủ để đáp ứng mọi yêu cầu về tổ chức dữ liệu không ?

**3)** Một ngôn ngữ lập trình có nên cho phép người sử dụng tự định nghĩa thêm các kiểu dữ liệu có cấu trúc? Giải thích và cho ví dụ.

**4)** Cấu trúc dữ liệu và cấu trúc lưu trữ khác nhau những điểm nào? Một cấu trúc dữ liệu có thể có nhiều cấu trúc lưu trữ được không? Ngược lại, một cấu trúc lưu trữ có thể tương ứng với nhiều cấu trúc dữ liệu được không ? Cho ví dụ minh hoạ.

**5)** Giả sử có một bảng giờ tàu cho biết thông tin về các chuyến tàu khác nhau của mạng đường sắt. Hãy biểu diễn các dữ liệu này bằng một cấu trúc dữ liệu thích hợp (file, array, struct ...) sao cho dễ dàng truy xuất giờ khởi hành, giờ đến của một chuyến tàu bất kỳ tại một nhà ga bất kỳ.

**6)** Theo anh / chị, thế nào là cấu trúc dữ liệu động, cấu trúc dữ liệu tĩnh? Cho ví dụ minh họa.

**7)** Hãy đánh giá độ phức tạp của đoạn code chạy trên C++ như sau:

|  |  |
| --- | --- |
| *1*  *2*  *3*  *4*  *5*  *6*  *7*  *8*  *9*  *10* | int Check (int a[], int n)  {  int i = 0;  while(true) {  if(a[i]<0 || i>=n)  return -1;  i++;  }  return i;  } |

**8)** Hãy đánh giá độ phức tạp của đoạn code chạy trên C++ như sau:

|  |  |
| --- | --- |
| *1*  *2*  *3*  *4*  *5*  *6*  *7*  *8*  *9*  *10* | bool duplicate = false;  for ( int i = 0; i < n; ++i ) {  for ( int j = 0; j < n; ++j )  if ( i != j && A[ i ] == A[ j ] ) {  duplicate = true;  break;  }  if ( duplicate )  break;  } |

**9)** Hãy đánh giá độ phức tạp của đoạn code chạy trên C++ như sau:

|  |  |
| --- | --- |
| *1*  *2*  *3*  *4*  *5*  *6*  *7*  *8*  *9*  *10*  *11*  *12*  *13* | void output(bool VertexName)  {  int tong=0;  for(int i=0; i<nT; i++)  {  if(VertexName)  cout<<endl<<"("<<vertex[T1[i]]<<","<<vertex[T2[i]]<<") = "<<wT[i];  else  cout<<endl<<"("<<T1[i]<<","<<T2[i]<<") = "<<wT[i];  tong+=wT[i];  }  cout<<"\n Tong = "<<tong;  } |

**10)** Hãy đánh giá độ phức tạp của đoạn code chạy trên C++ như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | void inputGraphFromText()  {  ifstream myfile ("matranke1.txt");  if (myfile.is\_open())  {  myfile>>n;  for(int i=0;i<n;i++)  myfile>>vertex[i];  for(int i=0;i<n;i++)  {  for(int j=0;j<n;j++)  myfile>>A[i][j];  }  }  } |

**11)** Hãy đếm số ***phép so sánh bằng*** trong đoạn code sau, từ đó đánh giá độ phức tạp của đoạn code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | int search(int a[], int n, int x)  {  int i = 0;  while (i < n && a[i]!= x) {  i++;  if (i == n)  return -1;  }  return i;  } |

**12)** Hãy đếm số ***phép gán*** trong đoạn code sau, từ đó đánh giá độ phức tạp của đoạn code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | void quickSort(int a[], int low, int high)  {  if (low < high)  {  int pi = partition(a, low, high);  quickSort(a, low, pi - 1);  quickSort(a, pi + 1, high);  }  } |

**CHƯƠNG 2: MỘT SỐ CTDL CƠ BẢN**

**1)** Anh / chị hãy:

(a) Định nghĩa DSLK kép DLIST với mỗi phần tử là các đối tượng PhanSo (gồm tử số và mẫu số)

(b) Viết các hàm cần thiết để hàm main() sau thực thi

|  |  |
| --- | --- |
| *1*  *2*  *3*  *4*  *5*  *6*  *7*  *8*  *9*  *10*  *11*  *12* | *Void main()*  *{*  *PhanSo ps;*  *DLIST L;*  *CreateList(L);*  *while(1) {*  *NhapPhanSo( ps );*  *if(ps.TuSo == 0 && ps.MauSo == 0 )*  *Break;*  *AddHead(L, ps);*  *}*  *}* |

**2)** Giả sử có danh sách liên kết (DSLK) có cấu trúc như sau:

|  |  |
| --- | --- |
| *1*  *2*  *3*  *4*  *5*  *6*  *7*  *8*  *9*  *10*  *11*  *12* | *typedef struct tagSINHVIEN {*  *int MSSV;*  *char TenSV[100];*  *float DTB;*  *} SINHVIEN;*    *typedef struct tagNODE {*  *SINHVIEN SV;*  *tagNODE \*pNext;*  *} NODE, \*PNODE;*    *PNODE pHead; // là biến toàn cục, biến này quản lý danh sách liên kết* |

Viết hàm thực hiện việc:

(a) Nhập từ bàn phím vào thông tin của 10 sinh viên.

Thông tin này được đưa vào DLSK theo phương pháp thêm vào cuối.

(b) In ra danh sách những sinh viên có ĐTB <5.

(c) Giải phóng danh sách liên kết

(d) Viết hàm main để gọi các hàm trên

**3)** Cho một danh sách liên kết đôi đã lưu thông tin về sản phẩm trong một công ty, bao gồm:

1.Mã sản phẩm (kiểu số nguyên)

2.Tên sản phẩm (kiểu chuỗi)

3.Chủng loại (bằng Giấy, bằng Kim loại, bằng Nhựa)

4.Năm sản xuất (kiểu số nguyên)

5.Số năm bảo hành (kiểu số nguyên)

Hai con trỏ First, Last đang trỏ đến phần tử đầu tiên và cuối cùng trong danh sách trên. Hãy thực hiện các yêu cầu sau :

(a) Viết hàm sắp xếp các sản phẩm theo mã sản phẩm giảm dần

(b) Viết hàm xóa các sản phẩm đã hết hạn bảo hành ra khỏi danh sách khi thỏa điều kiện : Năm sản xuất + Số năm bảo hành > Năm hiện tại

**4)** Trình bày khái niệm của các loại danh sách? Ưu, nhược điểm và ứng dụng của mỗi

loại danh sách?

**5)** Hãy đưa ra các cấu trúc dữ liệu để quản lý các loại danh sách vừa kể trên? Mỗi loại

bạn hãy chọn ra một cấu trúc dữ liệu mà theo bạn là hay nhất? Giải thích sự lựa

chọn đó?

**6)** Trình bày thuật toán và cài đặt tất cả các thao tác trên danh sách liên kết đơn trong

trường hợp quản lý bằng con trỏ đầu và cuối trong danh sách?

**7)** Trình bày thuật toán và cài đặt tất cả các thao tác trên danh sách liên kết đôi trong

trường hợp chỉ quản lý bằng con trỏ đầu trong danh sách?

**8)** Trình bày thuật toán và cài đặt tất cả các thao tác trên hàng đợi, ngăn xếp biểu diễn

bởi danh sách liên kết đôi trong hai trường hợp: Danh sách liên kết cùng chiều và

ngược chiều với hàng đợi, ngăn xếp?

**9)** Sử dụng Stack, viết chương trình nhập vào một số nguyên, không âm bất kỳ, sau

đó xuất ra màn hình số đảo ngược thứ tự các chữ số của số nhập vào.

**10)** Sử dụng Stack, viết chương trình chuyển đổi một số nguyên N trong hệ thập phân

(hệ 10) sang biểu diễn ở:

a. Hệ nhị phân (hệ 2)

b. Hệ thập lục phân (hệ 16)

**11)** Hãy sử dụng cấu trúc dữ liệu thích hợp để lưu trữ các ***số phức*** trong bộ nhớ trong của máy tính. Với cấu trúc dữ liệu này, hãy trình bày thuật toán và cài đặt chương trình thực hiện việc cộng, trừ, nhân, chia, so sánh các số phức này.

**CHƯƠNG 3: XẾP THỨ TỰ VÀ TÌM KIẾM**

***1)*** Trình bày giải thuật **tìm kiếm nhị phân**. Mô tả quá trình tìm giá trị x=8 trong mảng A={1,2,3,4,6,7,8,9} bằng phương pháp tìm kiếm nhị phân.

***2)*** Trình bày ngắn gọn ý tưởng thuật toán **Heap Sort**. Mô tả từng bước kết quả chạy thuật toán với mảng các số nguyên cho dưới đây để sắp xếp mảng tăng dần:

22 12 9 7 31 19 27

***3)*** Viết hàm cài đặt thuật toán sắp xếp **chọn trực tiếp** (Seclection sort).

**4)** Anh/ chị hãy:

(a) Trình bày các bước thuật toán **Chèn trực tiếp** để sắp xếp một mảng N số nguyên **giảm dần**.

(b) Trình bày các bước ( vẽ từng bước ) thực hiện sắp xếp theo thuật toán ở câu a đối với mảng 6 số nguyên có giá trị như sau: 31 ; 15 ; 37 ; 71 ; 20 ; 58

**5)** Anh/ chị hãy:

(a) Viết hàm cài đặt bằng ngôn ngữ C/C++ thuật toán Tìm kiếm **nhị phân** trên mảng số nguyên đã sắp xếp **giảm dần** N phần tử.

(b) Trình bày các bước (vẽ từng bước) theo hàm đã cài đặt ở câu a thực hiện tìm giá trị X=95 trong mảng 5 số nguyên có giá trị như sau: 90 ; 81 ; 65 ; 62 ; 42

(c) Gỉa sử cho mảng gồm N phần tử và phần tử X cần tìm cho trước (giả sử X có trong mảng), hãy cho biết số lần so sánh ít nhất và nhiều nhất giữa X và các phần tử trong mảng để tìm thấy phần tử X, cho ví dụ minh họa.

**6)** Anh/ chị hãy:

(a) Trình bày thuật toán **Quick Sort** để sắp xếp một mảng các số nguyên tăng dần.

(b) Cài đặt thuật toán trên bằng ngôn ngữ C++.

(c) Trình bày các bước (vẽ từng bước) thực hiện sắp xếp theo hàm đã cài đặt ở câu a đối với mảng sau: 19 11 31 15 37 17

**7)** Anh/ chị hãy:

(a) Trình bày ý tưởng thuật toán **Quick Sort** trên mảng. Thuật toán Quick Sort tốt hơn các thuật toán nào đã học ( so sánh ngắn gọn với ít nhất 2 thuật toán khác).

(b) Thể hiện từng bước kết quả chạy thuật toán trên ( Quick Sort ) với mảng sau:

22 12 9 7 31 19 2 27

**8)** Cho dãy số ban đầu như sau : 17 72 99 32 58 70 44 12 23 Hãy thực hiện các yêu cầu sau:

(a) Hãy trình bày các bước thực hiện thuật toán Bubble Sort để xếp thứ tự giảm dần.

(b) Áp dụng và vẽ hình từng bước thực hiện của thuật toán trên để sắp xếp dãy số theo thứ tự giảm dần (không cần lập trình)

**9)** Anh chị hãy:

(a) Trình bày ý tưởng thuật toán InterChange Sort trên mảng. Thuật toán InterChange Sort tốt hơn các thuật toán nào đã học (so sánh ngắn gọn với ít nhất 2 thuật toán khác).

(b) Thể hiện từng bước kết quả chạy thuật toán trên (InterChange Sort) với mảng sau:

22 12 9 7 31 19 2 27

**10)** Trong các thuật toán XẾP THỨ TỰ đã học, theo anh chị, thuật toán nào tối ưu nhất, tại sao? Hãy chọn ra 2 thuật toán trong các thuật toán đã học, so sánh 2 thuật toán đó.

**11)** Trong các thuật toán sắp xếp bạn thích nhất là thuật toán nào? Thuật toán nào bạn không thích nhất? Tại sao?

**12)** Thông tin về mỗi số hạng của một đa thức bậc n bao gồm: Hệ số – là một số thực, Bậc – là một số nguyên có giá trị từ 0 đến 100. Hãy định nghĩa cấu trúc dữ liệu để lưu trữ các đa thức trong bộ nhớ trong của máy tính. Với cấu trúc dữ liệu đã được định nghĩa, hãy vận dụng một thuật toán sắp xếp và cài đặt chương trình thực hiện việc sắp xếp các số hạng trong đa thức theo thứ tự tăng dần của các bậc.

**CHƯƠNG 4: CÂY NHỊ PHÂN TÌM KIẾM**

***1)***Trình bày khái niệm, đặc điểm và cấu trúc dữ liệu của các loại cây? So sánh với

danh sách liên kết?

**2)** Trình bày thuật toán và cài đặt tất cả các thao tác trên cây nhị phân tìm kiếm?

**3)** Trình bày thuật toán và cài đặt chương trình thực hiện các công việc sau trên cây nhị

phân:

a) Tính số nút lá của cây.

b) Tính số nút trung gian của cây.

c) Tính chiều dài đường đi tới một nút có khóa là K trên cây.

d) Cho biết mức (level) của một nút có khóa là K trên cây.

***4)*** Anh/ chị hãy:

(a) Định nghĩa cây nhị phân tìm kiếm (BST) có node là các số nguyên.

(b) Vẽ các bước cây NPTK được tạo lập từ tập số nguyên sau:

15 22 27 5 8 11 30 3 1 2

(b) Từ cây NPTK ở câu b, vẽ cây sau khi xóa tuần tự các số nguyên sau:

15 27 22 1

**5)** Cho dãy các ký tự như sau : A B C D E F W Z U T K

Hãy thực hiện các yêu cầu sau:

a. Hãy vẽ cây nhị phân tìm kiếm từ dãy ký tự trên

b. Bổ sung lần lượt các ký tự sau vào cây N, G, H , M , L để hình thành cây nhị phân tìm kiếm mới, vẽ hình cây khi thêm từng ký tự vào cây

c. Trình bày dãy kỹ tự kết qủa khi duyệt cây theo thứ tứ NRL, LRN , LNR

d. Vẽ hình cây khi xóa lần lượt các ký tự W, E, H, C

(Lưu ý: thứ tự trong chữ cái là thứ tự Alphabet)

**6)** Cho một cây nhị phân tìm kiếm T, mỗi node là một số nguyên.

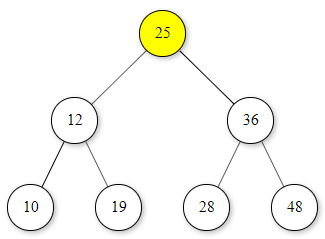
(a) Hãy vẽ cây nhị phân tìm kiếm T biết rằng khi duyệt cây theo thứ tự LRN như sau:

5 3 7 9 8 11 6 20 19 37 25 21 15 12

(b) Viết hàm đếm số node lá trên cây

(c) Viết hàm tìm phần tử có khóa là x trên cây.

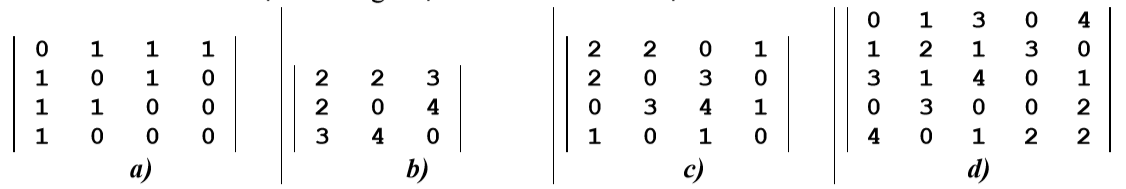
**7)** Cho cây NPTK như sau:



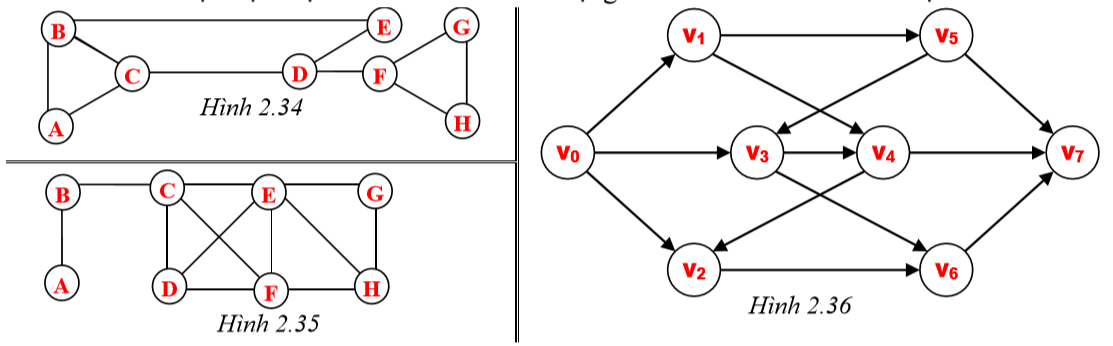
1. Bổ sung lần lượt các ký tự sau vào cây 52, 35, 28 , 17 , 66, 23, 11 để hình thành cây nhị phân tìm kiếm mới, vẽ hình cây khi thêm từng phần tử vào cây
2. Trình bày dãy kỹ tự kết qủa khi duyệt cây theo thứ tứ NRL, LRN , LNR
3. Vẽ hình cây khi xóa lần lượt các ký tự 52, 36, 12, 10

**CHƯƠNG 5: ĐỒ THỊ**

***1)***Vẽ đồ thị cho được biểu diễn bởi các ma trận sau:



***2)***Lần lượt thực hiện tìm kiếm theo chiều sâu và theo chiều rộng các đồ thị sau:



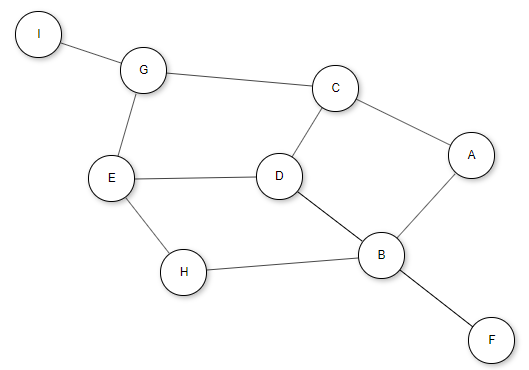
***3)*** Anh/ chị hãy:

(a) Trình bày công dụng của Stack và Queue? Trình bày thuật toán duyệt cây theo chiều sâu sử dụng đệ qui.

(b) Sử dụng Stack để khử đệ qui trong thuật toán duyệt cây theo chiều sâu, viết đoạn chương trình trên. Giả thiết là chúng ta đã có sẵn các hàm của stack (hàm Init, Pop, Push).

(c) Áp dụng Queue để viết đoạn chương trình tìm kiếm theo chiều rộng. Giả thiết là chúng ta đã có sẵn các hàm của queue (hàm Init, Pop, Push).

**4)** Cho đồ thị sau:



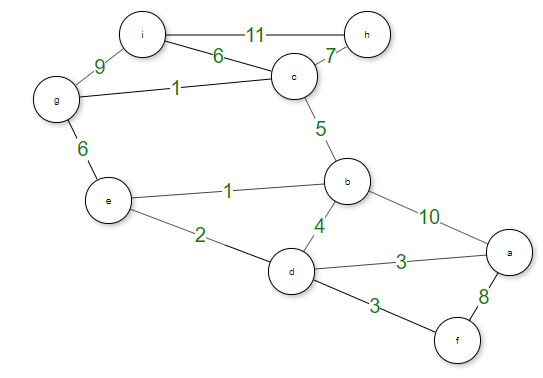
(a) Hãy cho biết Đồ thị trên là đồ thị gì? Tìm các chu trình trong đồ thị nếu có.

(b) Hãy xây dựng ma trận kề, danh sách kề cho đồ thị trên; Vẽ hình minh họa

(c) Hãy duyệt đồ thị theo BFS và DFS từ các đỉnh A, B và D;

(d) Hãy tìm đỉnh G của trên đồ thị bằng phương pháp duyệt DFS, giải thích minh họa.

**5)** Cho đồ thị sau



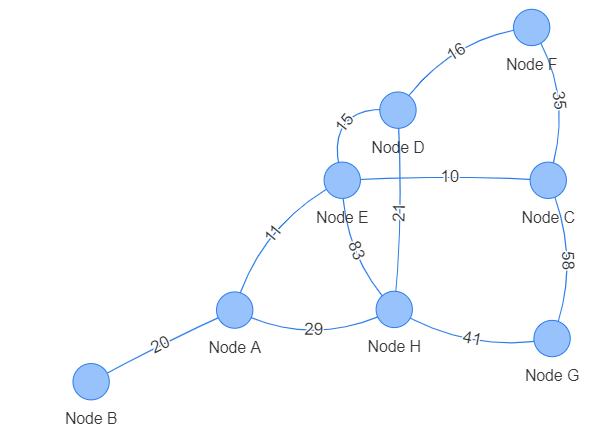
(a) Hãy cho biết Đồ thị trên là đồ thị gì? Tìm các chu trình trong đồ thị nếu có.

(b) Hãy xây dựng ma trận kề, danh sách kề cho đồ thị trên; Vẽ hình minh họa

(c) Hãy duyệt đồ thị theo BFS và DFS từ các đỉnh e, f và g;

(d) Hãy liệt kệ 5 đường đi khác nhau từ đỉnh a đến đỉnh e, sau đó tìm đường đi ngắn nhất từ a đến e.

**6)** Cho đồ thị như sau:



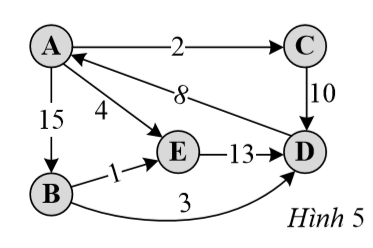
(a) Hãy cho biết Đồ thị trên là đồ thị gì? Tìm các chu trình trong đồ thị nếu có.

(b) Hãy xây dựng ma trận kề, danh sách kề cho đồ thị trên; Vẽ hình minh họa

(c) Hãy duyệt đồ thị theo BFS và DFS từ các đỉnh e, f và g;

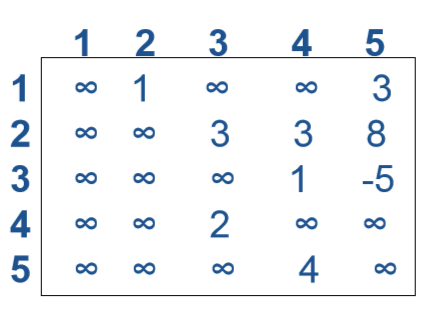
(d) Hãy liệt kệ 3 đường đi khác nhau từ đỉnh a đến đỉnh e, sau đó tìm đường đi ngắn nhất từ a đến e.

**7)** Cho đồ thị như hình sau



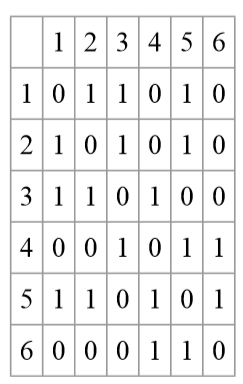
1. Hãy cho biết đồ thị trên là đồ thị gì? Tìm các chu trình trong đồ thị nếu có.
2. Hãy xác định tính liên thông của đồ thị trên.
3. Hãy xây dựng ma trận kề, danh sách kề cho đồ thị trên; Vẽ hình minh họa
4. Hãy bỏ các hướng của các cạnh (cung), hãy duyệt đồ thị, hãy duyệt theo BFS và DSF xuất phát từ các đỉnh A, C, và D;

**8)** Cho ma trận trọng số sau:



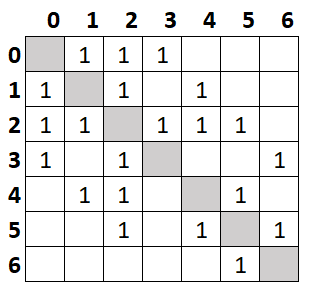
1. Hãy vẽ đồ thị được biểu diễn bởi ma trận trọng số trên và cho biết đồ thị là đồ thị gì? Tìm các chu trình trong đồ thị nếu có.
2. Hãy biểu diễn đồ thị trên bằng danh sách cạnh.

**9)** Cho ma trận vuông như sau:



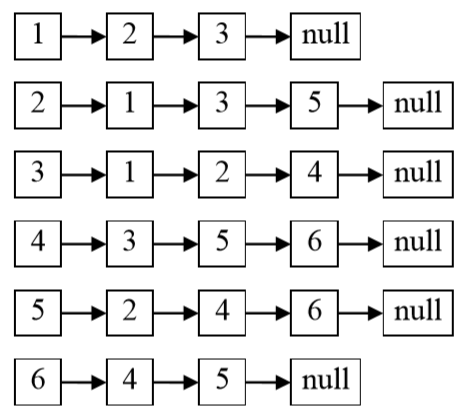
1. Hãy vẽ đồ thị được biểu diễn bởi ma trận trên và cho biết đồ thị là đồ thị gì? Tìm các chu trình trong đồ thị nếu có.
2. Hãy biểu diễn đồ thị trên bằng danh sách cạnh. Vẽ hình minh họa
3. Hãy duyệt đồ thị theo BFS và DSF xuất phát từ các đỉnh 1, 5, và 6;

**10)** Cho đồ thị được biểu diễn dưới dạng ma trận sau



1. Hãy vẽ đồ thị được biểu diễn bởi ma trận trên và cho biết đồ thị là đồ thị gì? Tìm các chu trình trong đồ thị nếu có.
2. Hãy biểu diễn đồ thị trên bằng danh sách cạnh. Vẽ hình minh họa
3. Hãy duyệt đồ thị theo BFS và DSF xuất phát từ các đỉnh 0, 4, và 2;

**11)** Cho danh sách cạnh của đồ thị như sau:



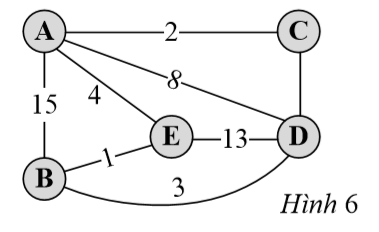
1. Hãy vẽ đồ thị được biểu diễn bởi danh sách cạnh trên, cho biết đây là đồ thị gì?
2. Hãy duyệt BFS và DFS từ đỉnh bất kỳ.

**CHƯƠNG 6: CÂY BAO TRÙM TỐI TIỂU**

**1)** Xây dựng đường ống nước

Có một trạm cấp nước và N điểm dân cư. Hãy trình bày ý tưởng xây dựng chương trình thiết kế tuyến đường ống nước cung cấp đến mọi nhà sao cho tổng chiều dài đường ống phải dùng là ít nhất. Giả sử rằng các đường ống chỉ được nối giữa hai điểm dân cư hoặc giữa trạm cấp nước với điểm dân cư.

**2)** Cho đồ thị như sau:

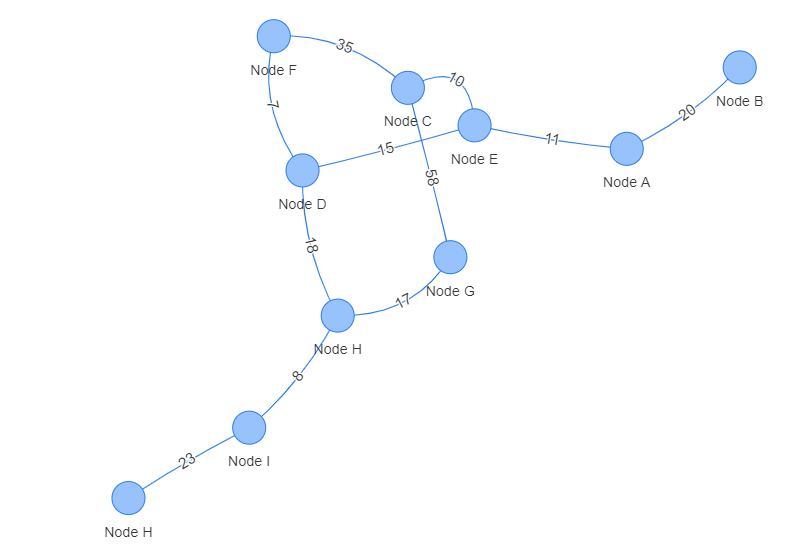


(a) Hãy tìm cây bao trùm tối tiểu bằng thuật toán PRIM, giải thích minh họa các bước thực hiện.

(b) Hãy tìm cây bao trùm tối tiểu bằng thuật toán KRUSKAL cải tiến, giải thích minh họa các bước thực hiện.

(c) Vẽ cây bao trùm tối tiểu tìm được.

**3)** Cho đồ thị như sau:

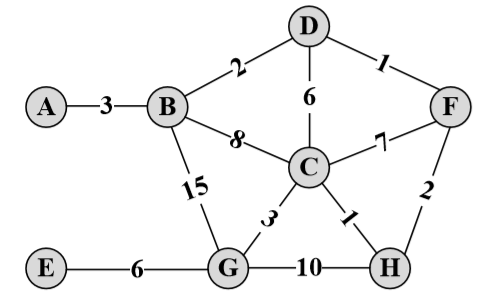


(a) Hãy tìm cây bao trùm tối tiểu bằng thuật toán PRIM, giải thích minh họa các bước thực hiện.

(b) Hãy tìm cây bao trùm tối tiểu bằng thuật toán KRUSKAL cải tiến, giải thích minh họa các bước thực hiện.

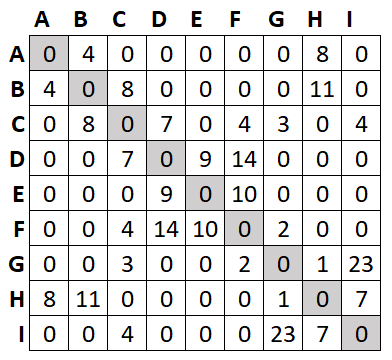
(c) Vẽ cây bao trùm tối tiểu tìm được

**4)** Cho đồ thị sau:



1. Thực hiện các bước của giải thuật PRIM để tìm cây khung nhỏ nhất của đồ thị.
2. Thực hiện các bước của giải thuật Kruskal để tìm cây khung nhỏ nhất của đồ thị.
3. Theo anh / chị, thì cây khung nhỏ nhất của đồ thị có phải là đường đi ngắn nhất giữa 2 đỉnh bất kỳ? Giải thích tại sao?

**5)** Cho ma trận sau:



1. Hãy vẽ đồ thị được biểu diễn bởi ma trận trên và cho biết đồ thị là đồ thị gì?
2. Thực hiện các bước của giải thuật PRIM để tìm cây khung nhỏ nhất của đồ thị.
3. Thực hiện các bước của giải thuật Kruskal để tìm cây khung nhỏ nhất của đồ thị

**6)** Tìm cây khung nhỏ nhất cho các đồ thị được biểu diễn dưới dạng ma trận sau:

|  |  |
| --- | --- |
| Đồ thị G1 | Đồ thị G2 |