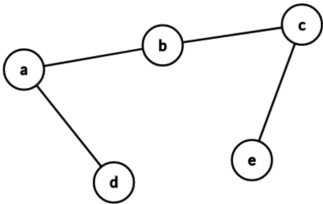


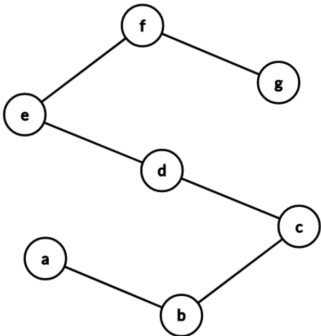
- 1 Mở đầu về cây
- 2 Các ứng dụng của cây
- 3 Các phương pháp duyệt cây
- 4 Cây khung

1. Xóa đi  $m - n + 1$  cạnh.

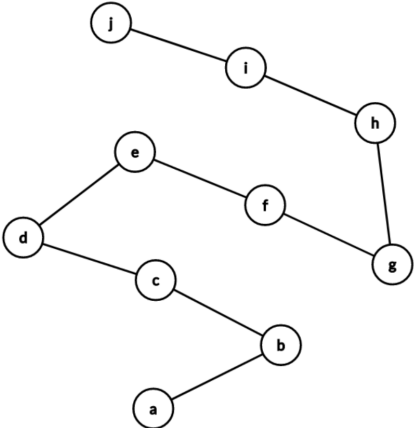
2.



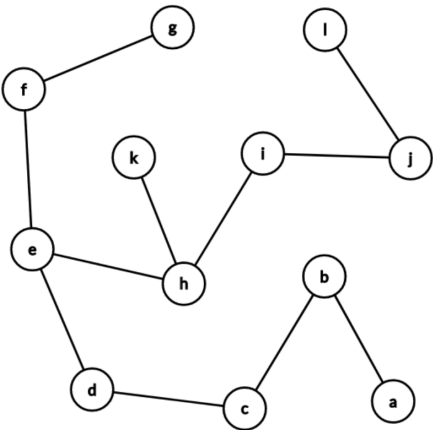
3.



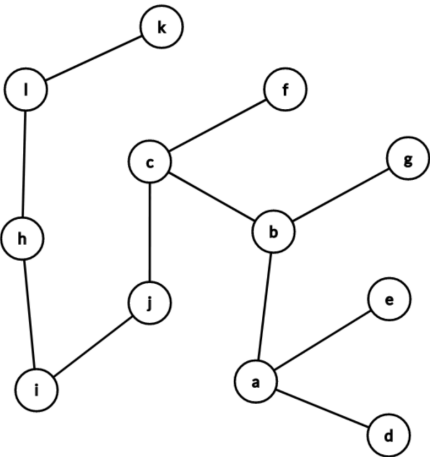
4.



5.

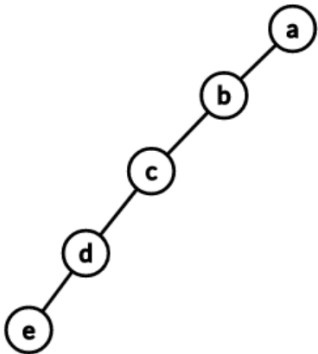


6.

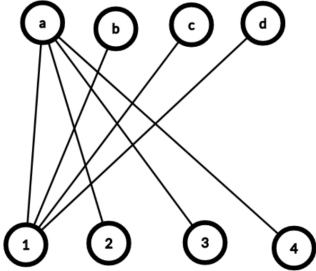


7.

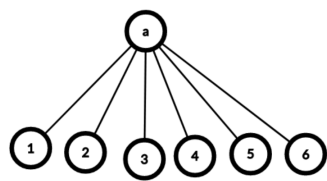
a)  $K_5$



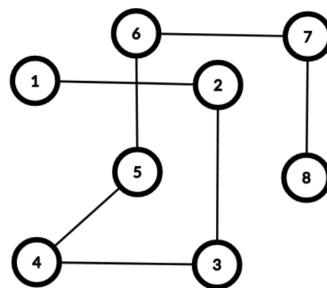
b)  $K_{4,4}$



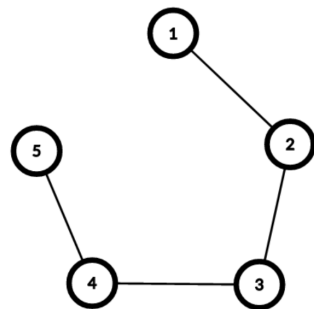
c)  $K_{1,6}$



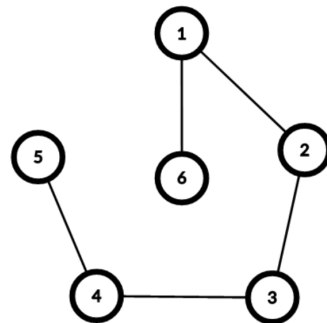
d)  $Q_3$



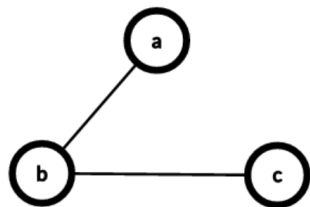
e)  $C_5$



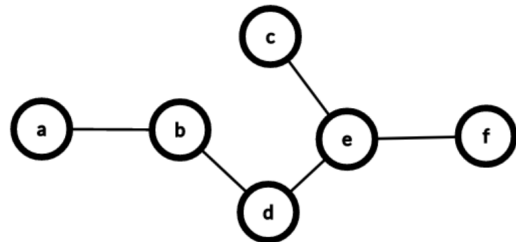
f)  $W_5$



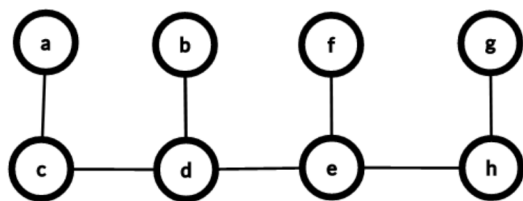
8.



10.



9.



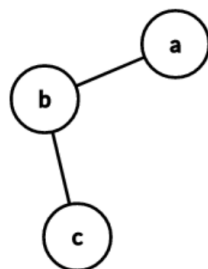
11\*.

- a)  $K_3$  : Số cây khung khác nhau là  $3^{3-2} = 3$ .  
c)  $K_{2,2}$  : Số cây khung khác nhau là  $2^1 \cdot 2^1 = 4$ .

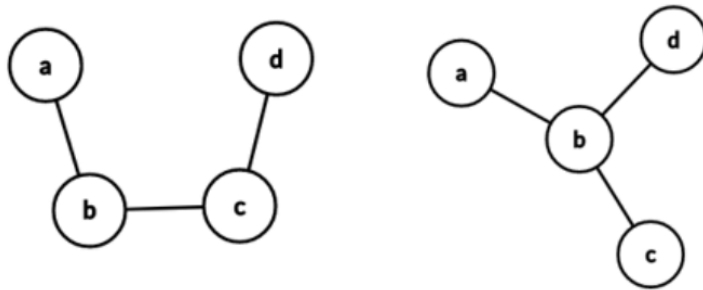
- b)  $K_4$  : Số cây khung khác nhau là  $4^{4-2} = 16$ .  
d)  $C_5$  : Số cây khung khác nhau là 5 (bỏ 1 cạnh trong số 5 cạnh).

12\*.

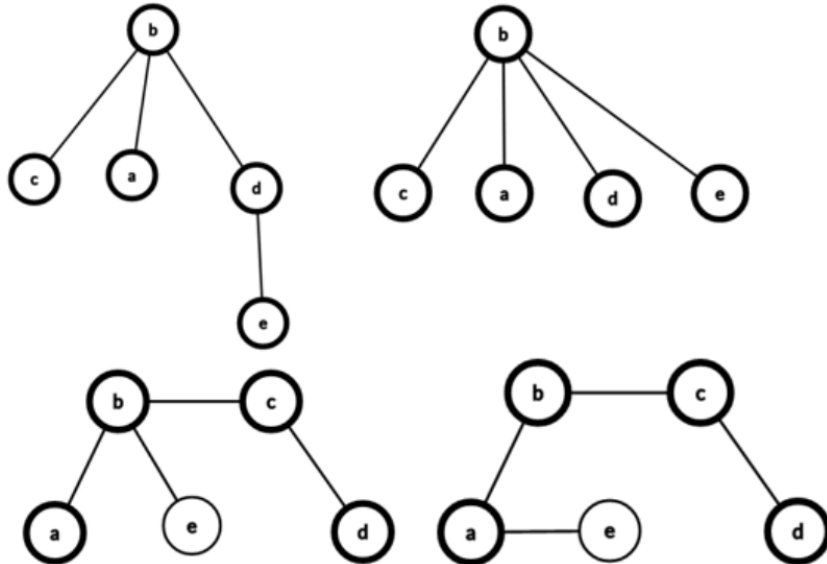
- a)  $K_3$  có 1 cây khung không đẳng cấu.



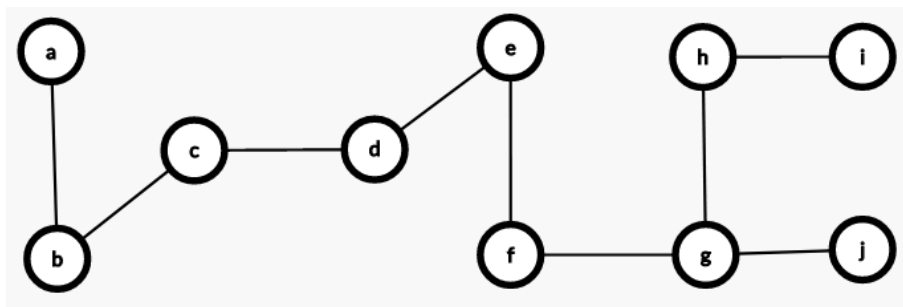
- b)  $K_4$  có 2 cây khung không đẳng cấu.



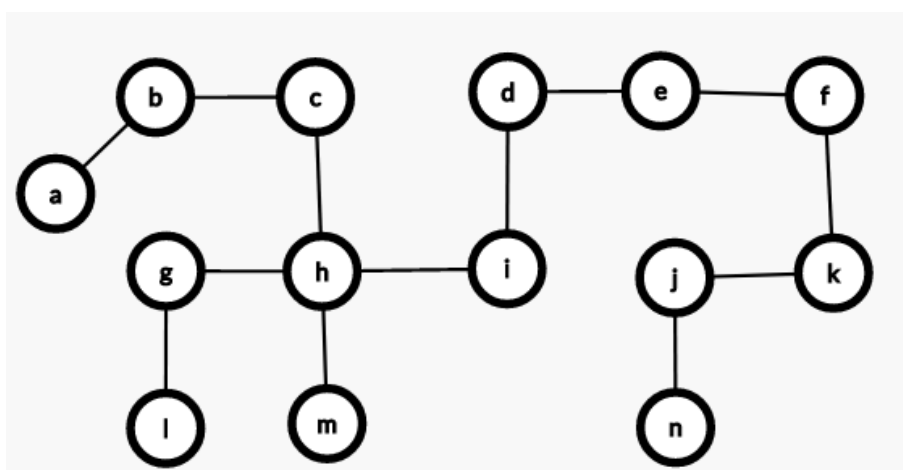
c)  $K_5$  có 4 cây khung không đẳng cấu.



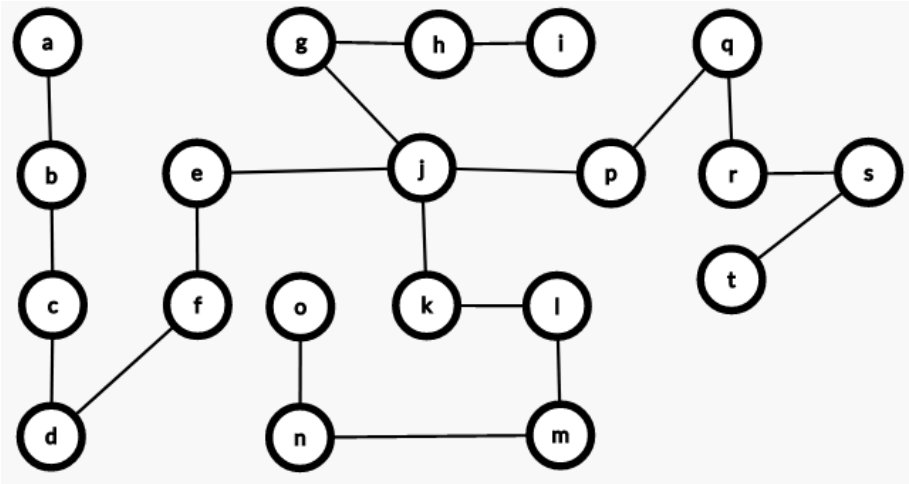
13.



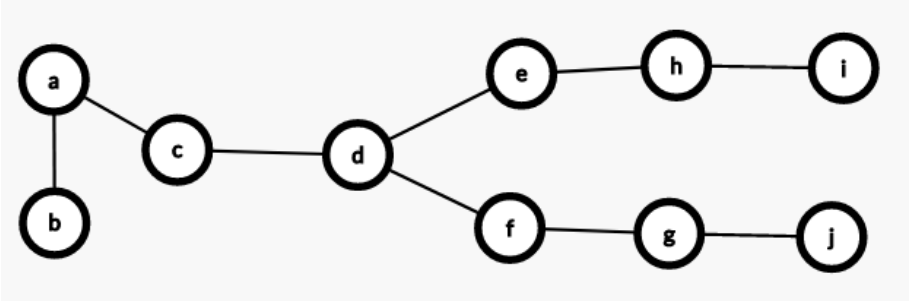
14.



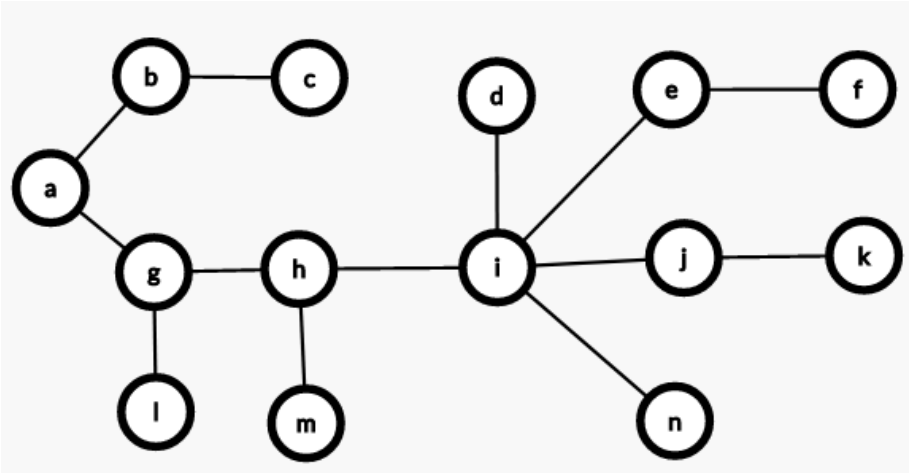
15.



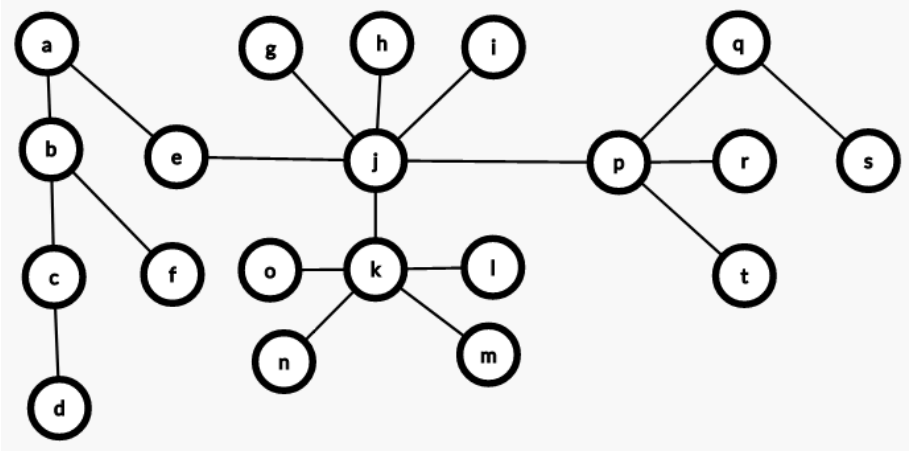
16.  
13)



14)

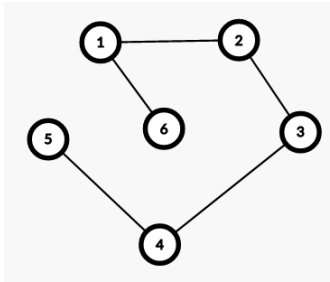


15)

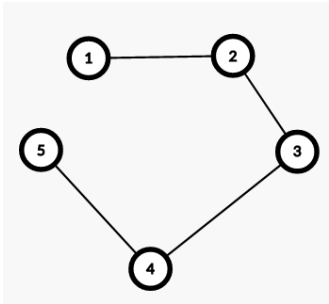


17.

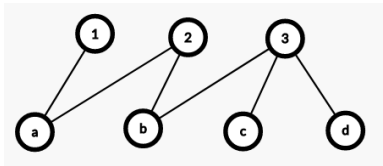
a)  $W_6$



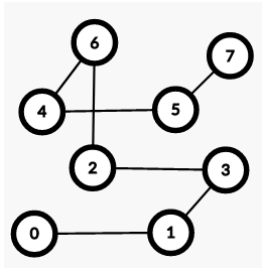
b)  $K_5$



c)  $K_{3,4}$

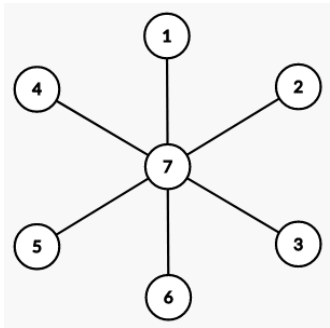


d)  $Q_3$

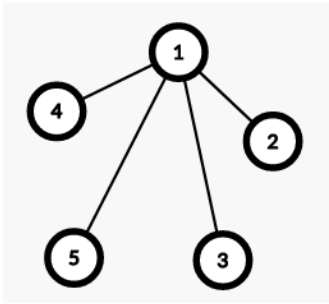


18.

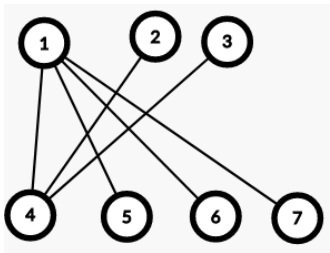
a)  $W_6$



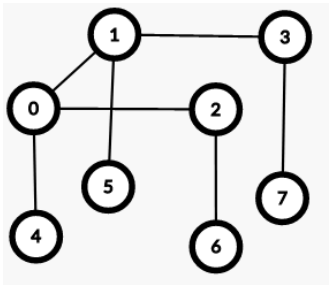
b)  $K_5$



c)  $K_{3,4}$

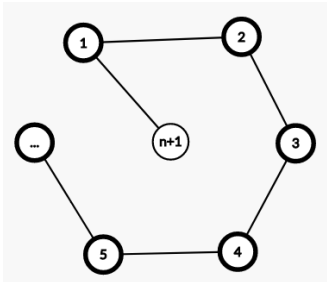


d)  $Q_3$

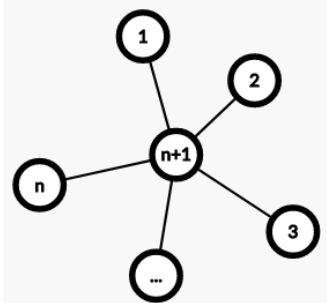


19.

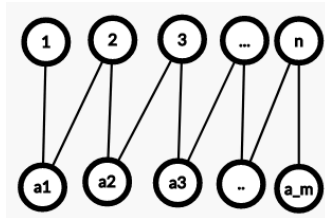
i) Tìm kiếm ưu tiên chiều sâu: Từ đỉnh  $n + 1$  sẽ DFS ra đỉnh số 1, bắt đầu kể từ đó đỉnh thứ  $i$  sẽ DFS trực tiếp ra đỉnh  $i + 1$  cho đến hết.



ii)



i) Tìm kiếm ưu tiên chiều sâu.



Tại mỗi thời điểm, đỉnh  $x$  có thể di chuyển đến đỉnh nhỏ nhất chưa được thăm ở phía bên kia, cứ như thế đồ thị sẽ có hình zig zag cho đến hết.

ii) Tìm kiếm ưu tiên chiều rộng

22.

- Thuật toán tìm kiếm ưu tiên chiều sâu: Xuất phát từ 1 đỉnh bất kỳ và tô màu cho đỉnh đó khác với đỉnh cha của nó khi DFS.
- Thuật toán tìm kiếm ưu tiên chiều rộng:

24. Khi 1 cạnh đơn của đồ thị liên thông cần phải có trong mọi cây khung, điều đó có nghĩa rằng nếu bỏ nó đi thì đồ thị chắc chắn bị chia làm hai phía, nếu không đã có thể dùng 1 cạnh khác thay thế vào cây khung rồi. Điều này có nghĩa rằng cạnh này là **khớp** của đồ thị.

25. Đơn đồ thị liên thông có đúng một cây khung là **cây**.

27\*. Khi BFS từ đỉnh  $v$  tại bước số 1, **mọi** đỉnh có  $d = 1$  đều đã được thăm, và đó là đường đi ngắn nhất từ đỉnh đó tới gốc  $v$ . Tương tự, tại bước số 2, tất cả các đỉnh có  $d = 2$  đều được thăm, chúng đã không thể được thăm tại bước 1. Như vậy, mỗi khi BFS tới 1 đỉnh  $u$  bất kỳ, tức  $u$  không thể nào được thăm ở các bước trước đó, nên  $d(u)_{\min}$  chính là độ sâu của  $u$  trong cây BFS gốc  $v$ .

29.

- a) Đặt tại  $(1, 1)$ ,  $(2, 3)$ , và  $(3, 5)$ .                      b) Đặt tại  $(1, 1)$ ,  $(2, 3)$ ,  $(3, 5)$ ,  $(4, 7)$ .  
c) Đặt tại  $(1, 1)$ ,  $(2, 3)$ ,  $(3, 5)$ ,  $(4, 7)$ ,  $(5, 2)$ ,  $(6, 4)$ .

30.

- a) Không có.                      b) 27, 24                      c) 8, 11, 24, 27

31.

- Tìm chu trình: bắt đầu từ 1 đỉnh bất kỳ.
- Tìm đường đi: Duyệt đỉnh khởi đầu từ đỉnh 1 đến đỉnh  $n$ .

Backtrack sinh hoán vị  $2^n$ , kiểm tra 2 đỉnh kề nhau có cạnh nối hay không.

39. Xét một cạnh nối từ  $u$  đến  $v$  không nằm trong  $T$ . Ta sẽ chứng minh  $u$  và  $v$  một trong hai chắc chắn phải là tổ tiên của đỉnh còn lại.

Trong quá trình DFS tới  $u$ , ta xét cạnh  $(u, v)$  thì  $v$  đã được thăm trước đó rồi. Nếu hàm đệ quy đến  $v$  đã kết thúc, thì hiển nhiên  $u$  đã phải được thêm vào cây khung với vai trò là một con cháu của  $v$  rồi. Còn nếu hàm đó chưa kết thúc, thì hiển nhiên  $u$  đang chính là một con cháu của  $v$  (mọi đỉnh DFS sau  $v$  khi  $v$  chưa xong đều thuộc cây con gốc  $v$ ).

Do đó không có chuyện  $u, v$  thuộc 2 cây con gốc khác nhau trong cây khung  $T$ , mà một trong hai là tổ tiên đỉnh còn lại.