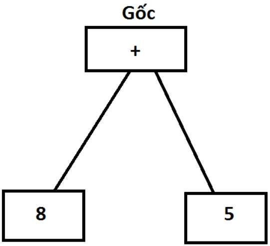
Cây nhị phân biểu thức

Xét cây như sau



Khi đó, theo phép duyệt
- Tiền thứ tự: + 8 5
- Hậu thứ tự: 8 5 +
- Trung thứ tự: 8 + 5

Cây nhị phân biểu thức

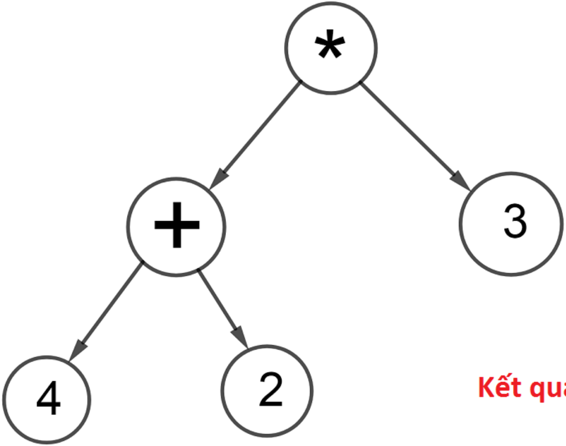
Định nghĩa

Cây nhị phân của biểu thức là cây nhị phân mà

- Mỗi biến số được biểu diễn bởi một lá.
- Mỗi đỉnh trong biểu diễn một phép toán với các thành tố là cây con tại đỉnh ấy.

Cây con bên trái và bên phải của một đỉnh trong biểu diễn cho biểu thức con, giá trị của chúng là thành tố mà ta áp dụng cho phép toán tại gốc của cây con.

Tính giá trị của biểu thức được biểu diễn bằng đồ thị sau

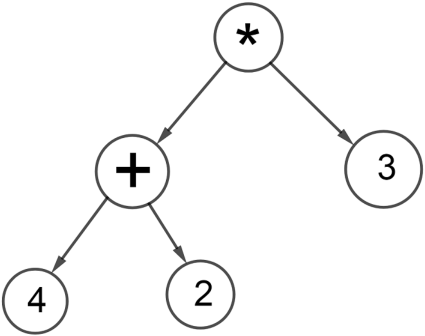


Kết quả: (4+2)*3=18

Định nghĩa

Ta gọi kết quả có được khi duyệt cây nhị phân của biểu thức theo phép duyệt

- Trung thứ tự là trung tố
- Tiền thứ tự là tiền tố và được gọi là ký pháp Ba Lan
- Hậu thứ tự là hậu tố và được gọi là ký pháp Ba Lan ngược

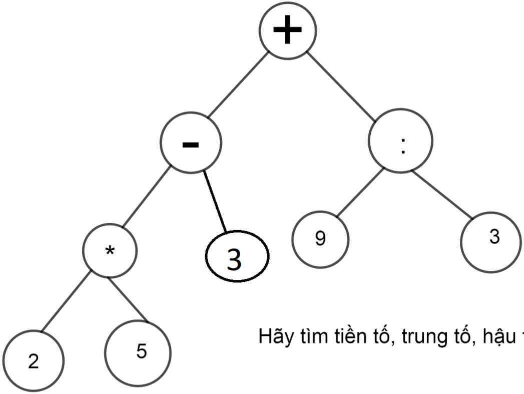


Khi đó

- Trung tố: 4 + 2 * 3
- Tiền tố: * + 4 2 3 Ký pháp Ba lan
- Hậu tố: 4 2 + 3 * Ký pháp Ba lan ngược

Ví dụ

Cho cây nhị phân T của biểu thức



Hãy tìm tiền tố, trung tố, hậu tố của G?

Ký pháp Ba Lan

Nhận xét. Để tính biểu thức khi có ký pháp Ba Lan ta tính từ phải sang trái: Bắt đầu từ bên phải, khi gặp một phép toán thì phép toán này được thực hiện cho 2 thành tố ngay bên phải nó, kết quả này là thành tố cho phép toán tiếp theo.

Ví dụ

Tính giá trị của ký pháp Ba Lan sau:

a) – * 2 / 8 4 3
b) ^ – * 3 3 * 4 2 5
c) + – ^ 3 2 ^ 2 3 / 6 – 4 2
d) *+ 3 + 3 ^ 3 + 3 3 3

Ký pháp Ba Lan ngược

Nhận xét. Để tính biểu thức khi có ký pháp Ba Lan ngược, ta tính từ bên trái, khi gặp một phép toán thì phép toán này được thực hiện cho 2 thành tố ngay bên trái nó, kết quả này là thành tố cho phép toán tiếp theo.

Ví dụ

Tính giá trị của ký pháp Ba Lan ngược sau:

a) 5 2 1−−3 1 4 ++*

b) 9 3 / 5 + 7 2 −*

c) 3 2* 2 ^ 5 3 − 8 4 / *−

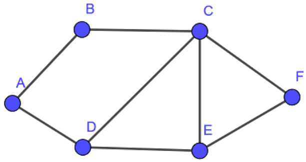
Cây khung của đồ thị

Định nghĩa

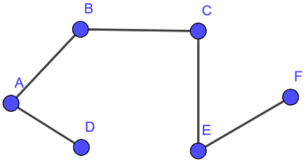
Một cây T được gọi là *cây khung* (hay cây tối đại, cây bao trùm) của đồ thị G=(V,E) nếu T là đồ thị con của G và chứa tất cả các đỉnh của G.

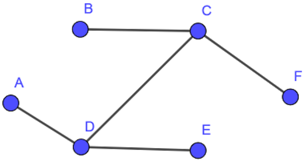
Ví dụ

Tìm cây khung của đồ thị sau



Đáp án. Một số cây khung của G





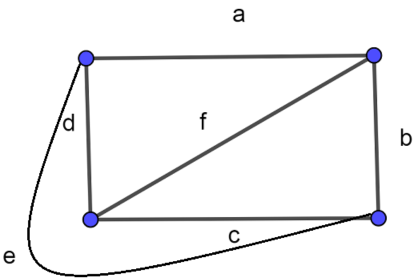
Nhận xét. Với 1 đồ thị cho trước, có thể có vài cây khung của đồ thị đó.

Định lý

Mọi đồ thị liên thông đều có cây khung

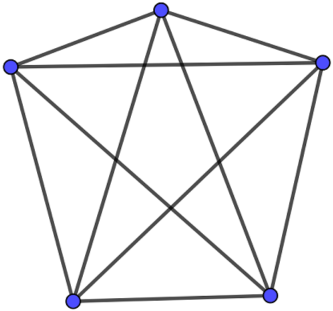
Định lý

Số cây khung của đồ thị K_n là n^{n-2}



$Số\ cây\ khung\quad 4^{4-2} = 16$

Ví dụ abc,bcd,cda,dab,abf...



Số cây khung$5^{5-2} = 125$

Tìm một cây khung của đồ thị

Bài toán: Cho G là đồ thị vô hướng liên thông, hãy tìm 1 cây khung của đồ thị G.

Để giải bài này ta dung 2 thuật toán sau

- Thuật toán tìm kiếm theo chiều rộng (BFS) Breadth-first search
- Thuật toán tìm kiếm theo chiều sâu (DFS) Depth-first search

Tìm kiếm theo chiều rộng (BFS)

Cho G là đồ thị liên thông với tập đỉnh $\{v_1, v_2, \ldots, v_n\}$

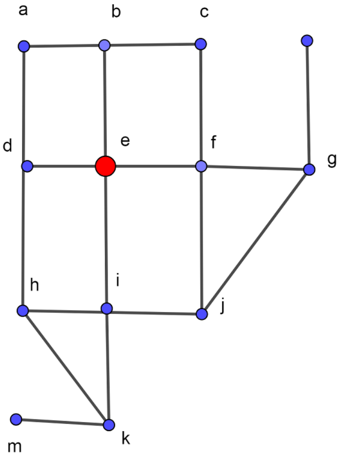
- Bước 0: thêm v_1 như là gốc của cây rỗng.
- Bước 1: thêm vào các đỉnh kề v_1 và các cạnh nối v_1 với chúng. Những đỉnh này là đỉnh mức 1 trong cây.
- Bước 2: đối với mọi đỉnh v mức 1, thêm vào các cạnh kề với v vào cây sao cho không tạo nên chu trình. Ta thu được các đỉnh mức 2.

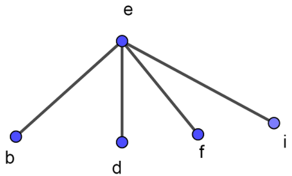
  ....

  Tiếp tục quá trình này cho tới khi tất cả các đỉnh của đồ thị được ghép vào cây. Cây T có được là cây khung của đồ thị.

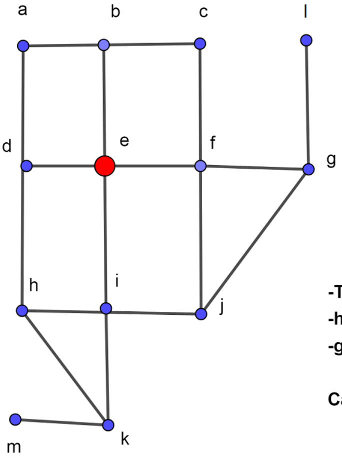
Ví dụ

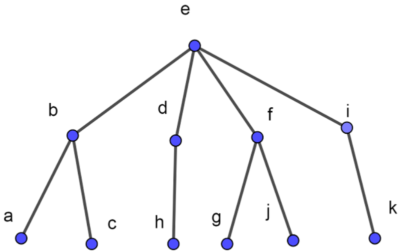
Tìm một cây khung của đồ thị G.





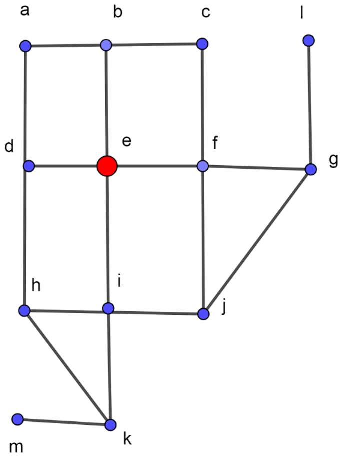
Chọn e làm gốc
Chọn các đỉnh kề với e
Các đỉnh mức 1 là: b, d, f, i

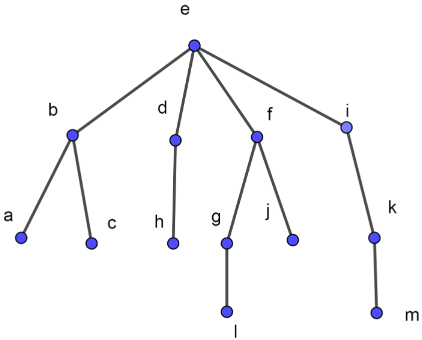




-Thêm a và c làm con của b
-h là con duy nhất của d
-g và j là con của f-k là con duy nhất của i

Các đỉnh mức 2 là: a, c, h, g, j, k





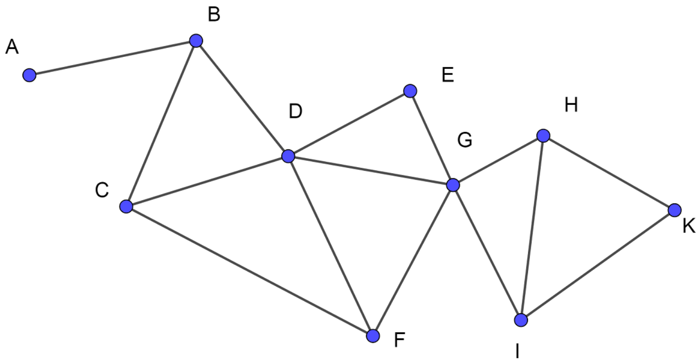
Cuối cùng thêm l và m là con của g và k tương ứng

Các đỉnh mức 3 là: l, m

Ta có được cây khung cần tìm

Ví dụ

Tìm cây khung của đồ thị bằng thuật toán BFS với D là đỉnh bắt đầu.



Tìm kiếm theo chiều sâu (DFS)

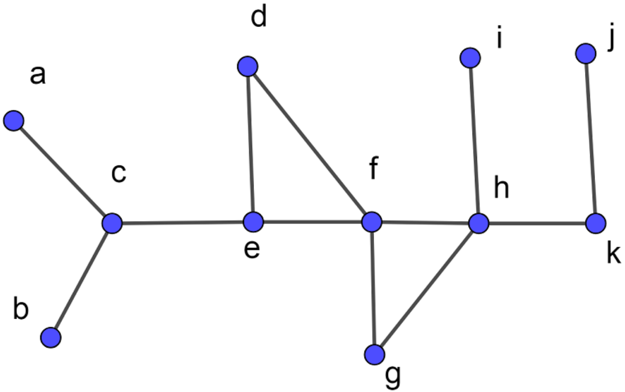
Cho G là đồ thị liên thông với tập đỉnh $\{v_1, v_2, \ldots, v_n\}$

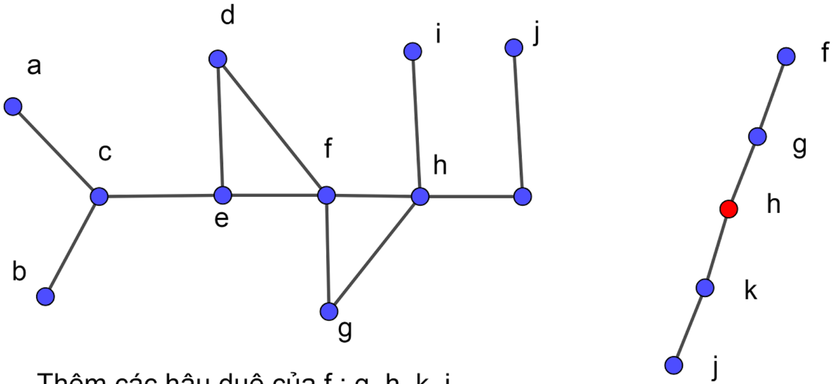
- Chọn một đỉnh tùy ý của đồ thị làm gốc.
- Xây dựng đường đi từ đỉnh này bằng cách lần lượt ghép các cạnh sao cho mỗi cạnh mới ghép sẽ nối đỉnh cuối cùng trên đường đi với một đỉnh còn chưa thuộc đường đi. Tiếp tục ghép thêm cạnh vào đường đi chừng nào không thể thêm được nữa.
- Nếu đường đi qua tất cả các đỉnh của đồ thị thì cây do đường đi này tạo nên là cây khung.

- Nếu chưa thì lùi lại đỉnh trước đỉnh cuối cùng của đường đi và xây dựng đường đi mới xuất phát từ đỉnh này đi qua các đỉnh còn chưa thuộc đường đi. Nếu điều đó không thể làm được thì lùi thêm một đỉnh nữa trên đường đi và thử xây dựng đường đi mới. Tiếp tục quá trình như vậy cho đến khi tất cả các đỉnh của đồ thị được ghép vào cây. Cây T có được là cây khung của đồ thị.

Ví dụ

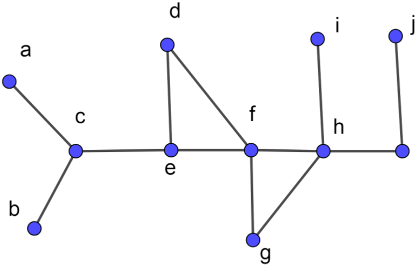
Tìm một cây khung của đồ thị với f là đỉnh gốc.

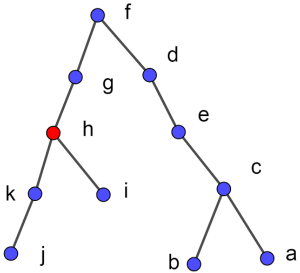




Thêm các hậu duệ của f : g, h, k, j

Lùi về k không thêm được cạnh nào, tiếp tục lùi về h





Thêm i làm con thứ hai của h và lùi về f.

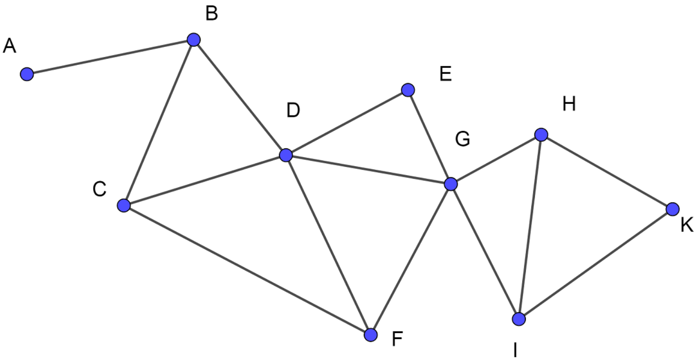
Lại thêm các hậu duệ của f : d, e, c, a

Lùi về c và thêm b làm con thứ hai của nó.

Cây thu được là cây khung của đồ thị đã cho

Ví dụ

Tìm một cây khung của đồ thị bằng thuật toán DFS với A là đỉnh bắt đầu.



Đồ thị có trọng số

Định nghĩa

Đồ thị G = (V,E) gọi là *đồ thị có trọng số* (hay chiều dài, trọng lượng) nếu mỗi cạnh e được gán với một số thực w(e). Ta gọi w(e) là trọng lượng của e.

- *Độ dài* của đường đi từ u đến v bằng tổng trọng lượng các cạnh mà đường đi qua.
- *Trọng lượng* của một cây T của G bằng với tổng trọng lượng các cạnh trong cây
- *Cây khung ngắn nhất* là cây khung có trọng lượng nhỏ nhất của G.

Ma trận khoảng cách (trọng số)

Định nghĩa

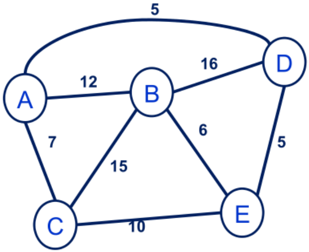
Cho G = (V, E), $V = \{v_1, v_2, \ldots, v_n\}$ là đơn đồ thị có trọng số. *Ma trận khoảng cách* của G là ma trận $D = (d_i j)$ được xác định như sau:

$$d_{ij} = \begin{cases} 0 & \text{khi } i = j \\ w(v_i v_j) & \text{khi } v_i v_j \in E \\ \infty & \text{khi } v_i v_j \notin E \end{cases}$$

Ma trận khoảng cách

Ví dụ

Tìm ma trận khoảng cách của đồ thị sau



$$\begin{bmatrix} 0 & 12 & 7 & 5 & \infty \\ 12 & 0 & 15 & 16 & 6 \\ 7 & 15 & 0 & \infty & 10 \\ 5 & 16 & \infty & 0 & 5 \\ \infty & 6 & 10 & 5 & 0 \end{bmatrix}$$

Thuật toán tìm cây khung ngắn nhất

Có nhiều thuật toán xây dựng cây khung ngắn nhất:

- Thuật toán Boruvka
- Thuật toán Kruskal
- Thuật toán Jarnik – Prim
- Thuật toán Chazelle

  ....

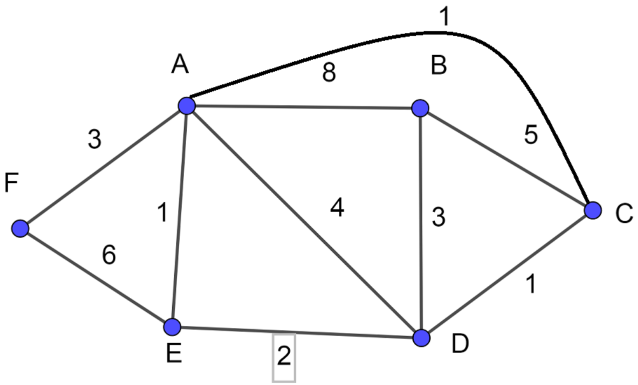
Thuật toán Kruskal

Input: Đồ thị G=(X, E) liên thông, X gồm n đỉnh.
Output: Cây khung ngắn nhất T=(V, U) của G
Bước 1. Sắp xếp các cạnh trong G tăng dần theo trọng lượng; khởi tạo T := $\emptyset$.
Bước 2. Lần lượt lấy từng cạnh e thuộc danh sách đã sắp xếp. Nếu T+$\{e\}$ không chứa chu trình thì thêm e vào T: T := $T + \{e\}$.
Bước 3. Nếu T đủ n-1 cạnh thì dừng; ngược lại, lặp bước 2.

Thuật toán Kruskal

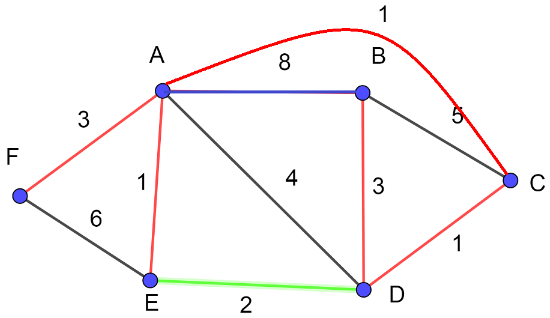
Ví dụ

Tìm cây khung ngắn nhất của đồ thị sau



Giải. Sắp xếp các cạnh

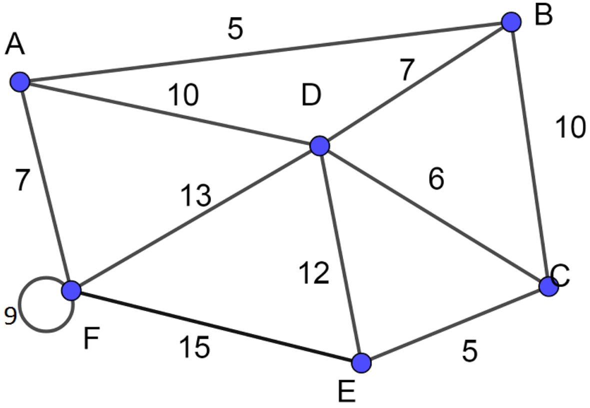
AC(1) - AE(1) - CD(1) - DE(2) - AF(3) - BD(3) - AD(4) - BC(5) - BF(6) - AB(8)



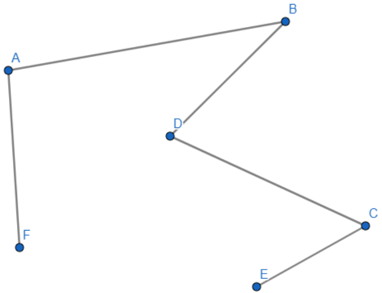
Như vậy T = AC, AE, CD, AF, BD là khung ngắn nhất với trọng lượng: 9

Ví dụ

Tìm cây khung ngắn nhất của đồ thị sau



Sắp xếp các cạnh $AB(5) - EC(5) - DC(6) - DB(7) - AF(7) - AD(10) - BC(10) - DE(12) - DF(13) - EF(15)$
Đặt $T = \emptyset$.
Bổ sung AB, EC, DC, DB, AF vào T. Vì đồ thị có 6 đỉnh nên thuật toán dừng lại khi ta đã có 5 cạnh. Do đó cây khung nhỏ nhất là AB, EC, DC, DB, AF, trọng lượng: 30.



Thuật toán Prim

Input: Đồ thị liên thông G=(X, E), X gồm n đỉnh

Output: Cây khung ngắn nhất T=(V, U) của G

Bước 1. Chọn tùy ý $v \in X$ và khởi tạo $V := \{v\}; U := \emptyset;$

Bước 2. Chọn cạnh e có trọng lượng nhỏ nhất trong các cạnh wv mà $w \in X \setminus V$ và $v \in V$

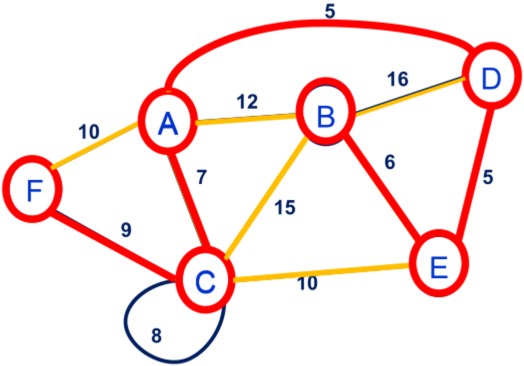
Bước 3. $V := V \cup \{w\}; U := U \cup \{e\}$

Bước 4. Nếu U đủ n-1 cạnh thì dừng, ngược lại lặp từ bước 2.

Thuật toán Prim

Ví dụ

Tìm cây khung ngắn nhất của đồ thị sau



V = {F, C, A, D, E, B}

U = {FC, CA, AD, DE, EB}

Trọng lượng: 32