§6. CÂY (TREE)

6.1. ĐỊNH NGHĨA

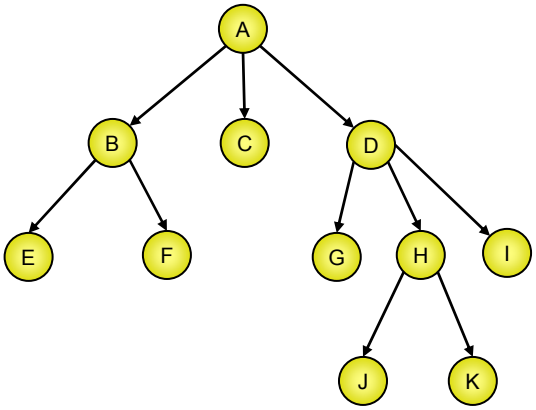
Cấu trúc dữ liệu trừu tượng ta quan tâm tới trong mục này là cấu trúc cây. Cây là một cấu trúc dữ liệu gồm một tập hữu hạn các nút, giữa các nút có một quan hệ phân cấp gọi là quan hệ "cha – con". Có một nút đặc biệt gọi là gốc (root).

Có thể định nghĩa cây bằng các đệ quy như sau:

- Mỗi nút là một cây, nút đó cũng là gốc của cây ấy
- Nếu n là một nút và $n_1, n_2, \ldots, n_k$ lần lượt là gốc của các cây $T_1, T_2, \ldots, T_k$; các cây này đôi một không có nút chung. Thì nếu cho nút n trở thành cha của các nút $n_1, n_2, \ldots, n_k$ ta sẽ được một cây mới T. Cây này có nút n là gốc còn các cây $T_1, T_2, \ldots, T_k$ trở thành các cây con (subtree) của gốc.

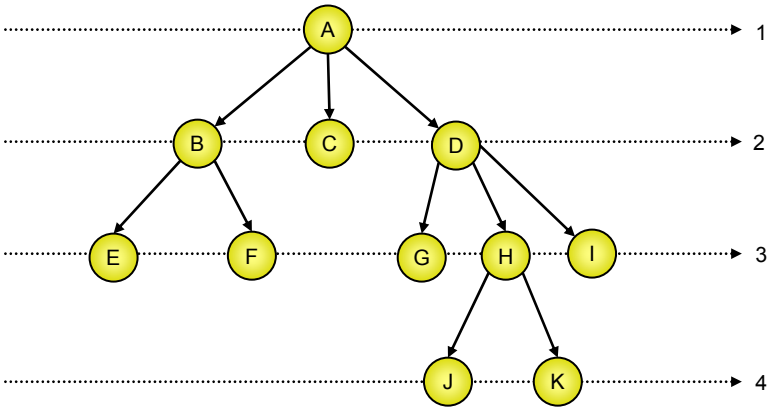
Để tiện, người ta còn cho phép tồn tại một cây không có nút nào mà ta gọi là cây rỗng (null tree).

Xét cây trong Hình 16:



Hình 16: Cây

- Quan hệ cha-con được định nghĩa theo hướng mũi tên. Ví dụ A là cha của B, C, D, còn G, H, I là con của D.
- Số các con của một nút được gọi là **cấp của nút** đó, ví dụ cấp của A là 3, cấp của B là 2, cấp của C là 0.
- Nút có cấp bằng 0 được gọi là **nút lá** (leaf) hay nút tận cùng: các nút E, F, C, G, J, K và I là các nút là. Những nút không phải là lá được gọi là **nút nhánh** (branch)
- Cấp cao nhất của một nút trên cây gọi là **cấp của cây** đó, cây ở Hình 16 là cây cấp 3.
- Gốc của cây người ta gán cho số mức là 1, nếu nút cha có mức là i thì nút con sẽ có mức là i + 1. Mức của cây trong Hình 16 được chỉ ra trong Hình 17:

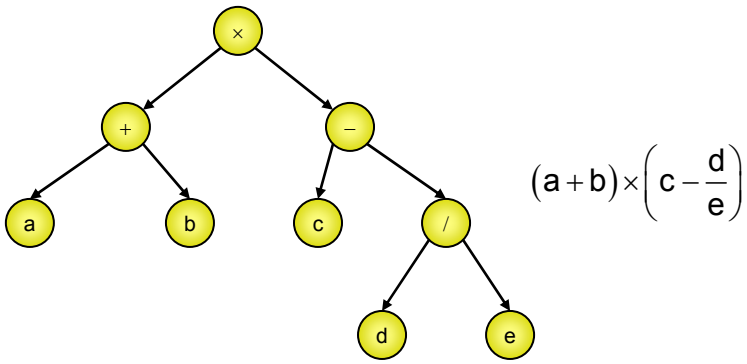


Hình 17: Mức của các nút trên cây

- **Chiều cao** (height) hay **chiều sâu** (depth) của một cây là số mức lớn nhất của nút có trên cây đó. Cây ở trên có chiều cao là 4
- Một tập hợp các cây phân biệt được gọi là **rừng** (forest), một cây cũng là một rừng. Nếu bỏ nút gốc trên cây thì sẽ tạo thành một rừng các cây con.

Ví dụ:

- Mục lục của một cuốn sách với phần, chương, bài, mục v.v… có cấu trúc của cây
- Cấu trúc thư mục trên đĩa cũng có cấu trúc cây, thư mục gốc có thể coi là gốc của cây đó với các cây con là các thư mục con và tệp nằm trên thư mục gốc.
- Gia phả của một họ tộc cũng có cấu trúc cây.
- Một biểu thức số học gồm các phép toán cộng, trừ, nhân, chia cũng có thể lưu trữ trong một cây mà các toán hạng được lưu trữ ở các nút lá, các toán tử được lưu trữ ở các nút nhánh, mỗi nhánh là một biểu thức con.



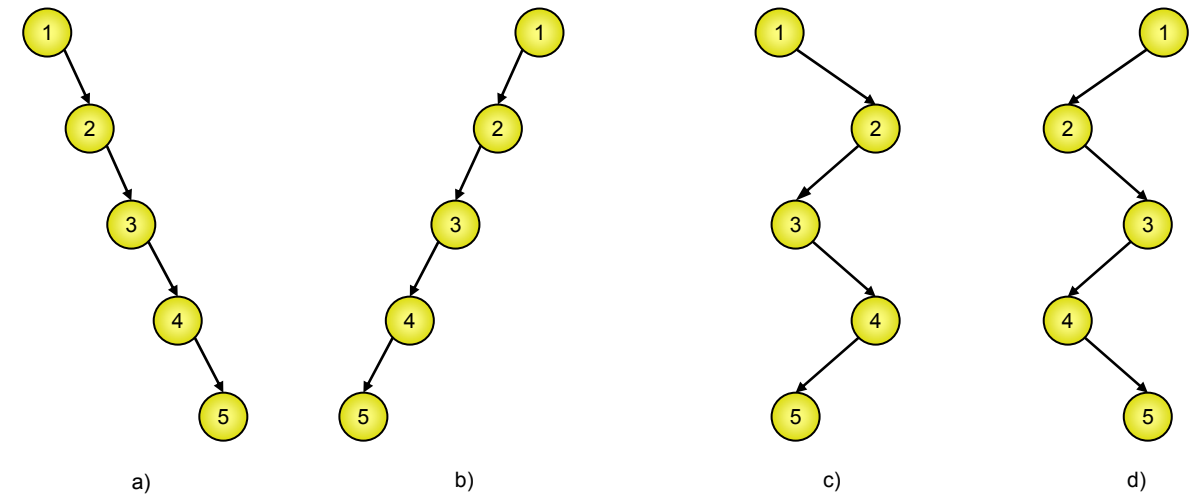
Hình 18: Cây biểu diễn biểu thức

6.2. CÂY NHỊ PHÂN (BINARY TREE)

Cây nhị phân là một dạng quan trọng của cấu trúc cây. Nó có đặc điểm là mọi nút trên cây chỉ có tối đa hai nhánh con. Với một nút thì người ta cũng phân biệt cây con trái và cây con phải của nút đó. Cây nhị phân là cây có tính đến thứ tự của các nhánh con.

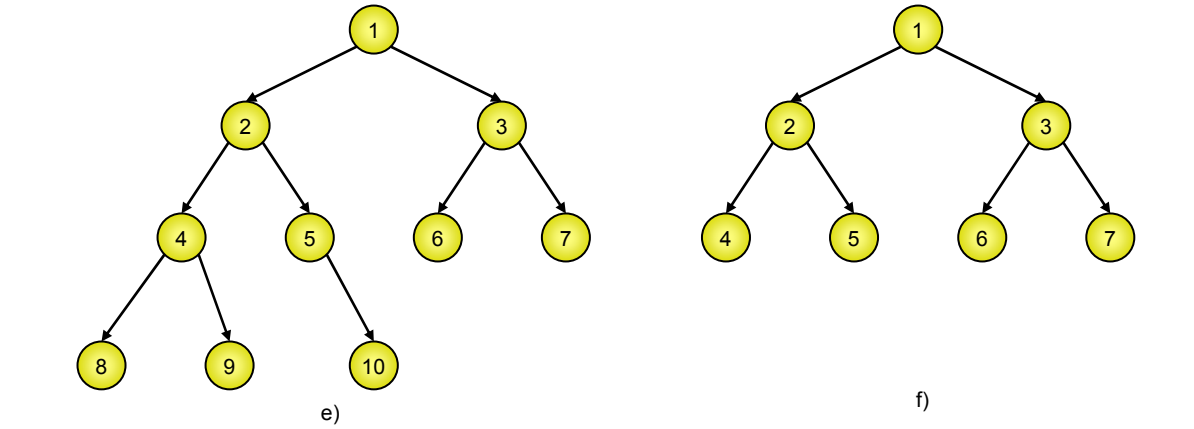
Cần chú ý tới một số dạng đặc biệt của cây nhị phân

Các cây nhị phân trong Hình 19 được gọi là **cây nhị phân suy biến** (degenerate binary tree), các nút không phải là lá chỉ có một nhánh con. Cây a) được gọi là cây lệch phải, cây b) được gọi là cây lệch trái, cây c) và d) được gọi là cây zíc-zắc.



Hình 19: Các dạng cây nhị phân suy biến

Các cây trong Hình 20 được gọi là **cây nhị phân hoàn chỉnh** (complete binary tree): Nếu chiều cao của cây là h thì mọi nút có mức < h - 1 đều có đúng 2 nút con. Còn nếu mọi nút có mức ≤ h - 1 đều có đúng 2 nút con như trường hợp cây f) thì cây đó được gọi là **cây nhị phân đầy đủ** (full binary tree). Cây nhị phân đầy đủ là trường hợp riêng của cây nhị phân hoàn chỉnh.



Hình 20: Cây nhị phân hoàn chỉnh và cây nhị phân đầy đủ

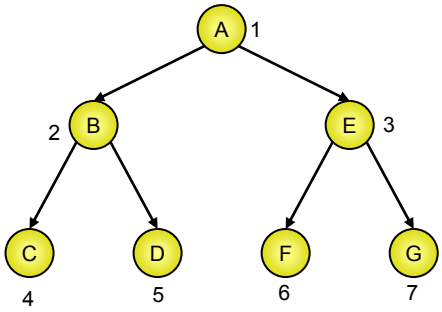
Ta có thể thấy ngay những tính chất sau bằng phép chứng minh quy nạp:

- Trong các cây nhị phân có cùng số lượng nút như nhau thì cây nhị phân suy biến có chiều cao lớn nhất, còn cây nhị phân hoàn chỉnh thì có chiều cao nhỏ nhất.
- Số lượng tối đa các nút trên mức i của cây nhị phân là 2^{i-1}, tối thiểu là 1 ($i \geq 1$).
- Số lượng tối đa các nút trên một cây nhị phân có chiều cao h là 2^h-1, tối thiểu là h ($h \geq 1$).
- Cây nhị phân hoàn chỉnh có n nút thì chiều cao của nó là $h = \lfloor \lg(n) \rfloor + 1$.

6.3. BIỂU DIỄN CÂY NHỊ PHÂN

6.3.1. Biểu diễn bằng mảng

Nếu có một cây nhị phân đầy đủ, ta có thể dễ dàng đánh số cho các nút trên cây đó theo thứ tự lần lượt từ mức 1 trở đi, hết mức này đến mức khác và từ trái sang phải đối với các nút ở mỗi mức.



Hình 21: Đánh số các nút của cây nhị phân đầy đủ để biểu diễn bằng mảng

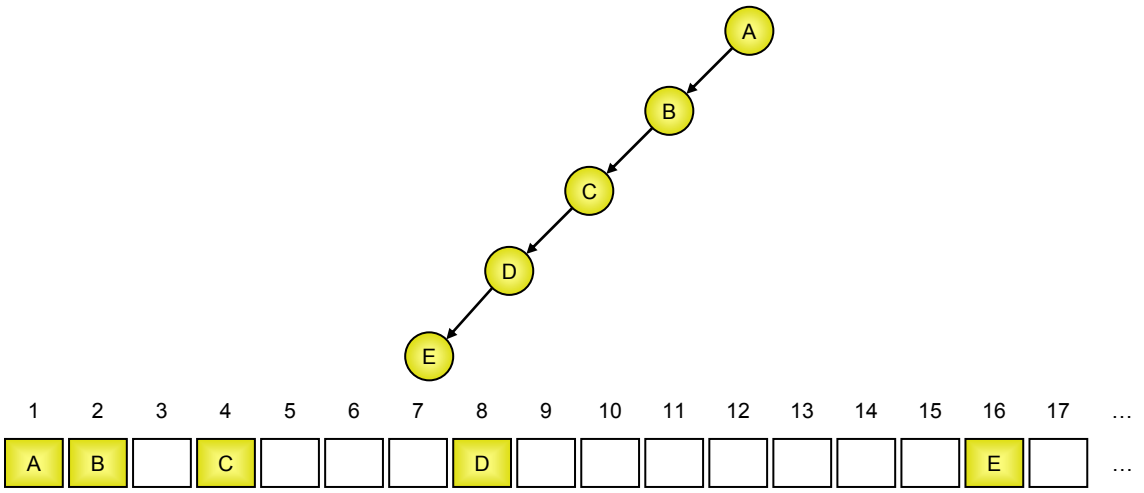
Với cách đánh số này, con của nút thứ i sẽ là các nút thứ 2i và 2i + 1. Cha của nút thứ i là nút thứ $\left\lfloor \frac{i}{2} \right\rfloor$. Từ đó có thể lưu trữ cây bằng một mảng T trong đó nút thứ i của cây được lưu trữ bằng phần tử T[i].

Với cây nhị phân đầy đủ ở Hình 21 thì khi lưu trữ bằng mảng, ta sẽ được mảng:

1	2	3	4	5	6	7
A	B	E	C	D	F	G

Trong trường hợp cây nhị phân không đầy đủ, ta có thể thêm vào một số nút giả để được cây nhị phân đầy đủ, khi biểu diễn bằng mảng thì những phần tử tương ứng với các nút giả sẽ được gán một giá trị đặc biệt. Một giải pháp khác là dùng thêm một mảng phụ để đánh dấu những nút nào là nút giả. Chính vì lý do này nên việc biểu diễn cây nhị phân không đầy đủ bằng mảng sẽ gặp phải sự lãng phí bộ nhớ nếu phải thêm nhiều nút giả cho được cây nhị phân đầy đủ.

Ví dụ với cây lệch trái, ta phải dùng một mảng 31 phần tử để lưu cây nhị phân chỉ gồm 5 nút

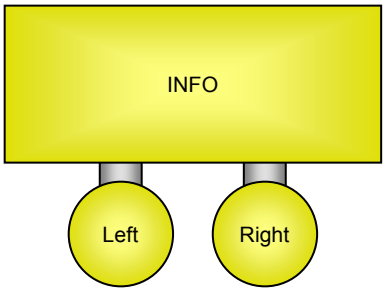


Hình 22: Nhược điểm của phương pháp biểu diễn cây nhị phân bằng mảng

6.3.2. Biểu diễn bằng cấu trúc liên kết.

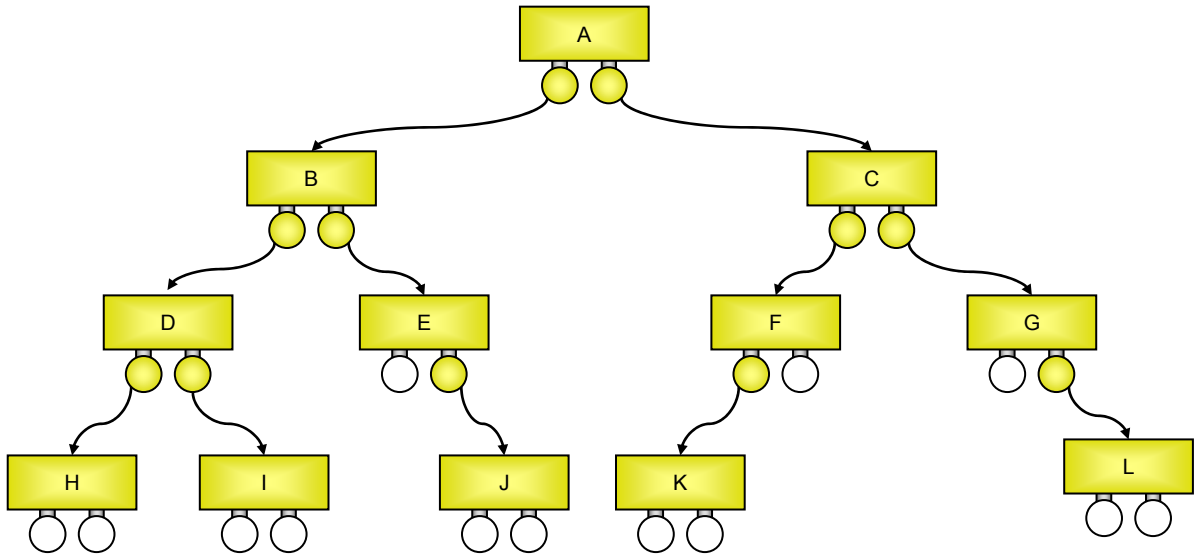
Khi biểu diễn cây nhị phân bằng cấu trúc liên kết, mỗi nút của cây là một bản ghi (record) gồm 3 trường:

- Trường Info: Chứa giá trị lưu tại nút đó
- Trường Left: Chứa liên kết (con trỏ) tới nút con trái, tức là chứa một thông tin đủ để biết nút con trái của nút đó là nút nào, trong trường hợp không có nút con trái, trường này được gán một giá trị đặc biệt.
- Trường Right: Chứa liên kết (con trỏ) tới nút con phải, tức là chứa một thông tin đủ để biết nút con phải của nút đó là nút nào, trong trường hợp không có nút con phải, trường này được gán một giá trị đặc biệt.



Hình 23: Cấu trúc nút của cây nhị phân

Đối với cây ta chỉ cần phải quan tâm giữ lại nút gốc, bởi từ nút gốc, đi theo các hướng liên kết Left, Right ta có thể duyệt mọi nút khác.



Hình 24: Biểu diễn cây nhị phân bằng cấu trúc liên kết

6.4. PHÉP DUYỆT CÂY NHỊ PHÂN

Phép xử lý các nút trên cây mà ta gọi chung là phép thăm (Visit) các nút một cách hệ thống sao cho mỗi nút chỉ được thăm một lần gọi là phép duyệt cây.

Giả sử rằng nếu như một nút không có nút con trái (hoặc nút con phải) thì liên kết Left (Right) của nút đó được liên kết thẳng tới một nút đặc biệt mà ta gọi là NIL (hay NULL), nếu cây rỗng thì nút gốc của cây đó cũng được gán bằng NIL. Khi đó có ba cách duyệt cây hay được sử dụng:

6.4.1. Duyệt theo thứ tự trước (preorder traversal)

Trong phép duyệt theo thứ tự trước thì giá trị trong mỗi nút bất kỳ sẽ được liệt kê trước giá trị lưu trong hai nút con của nó, có thể mô tả bằng thủ tục đệ quy sau:

```
procedure Visit(N); {Duyệt nhánh cây nhận N là nút gốc của nhánh đó}
begin
  if N ≠ nil then
    begin
      <Output trường Info của nút N>
      Visit(Nút con trái của N);
      Visit(Nút con phải của N);
    end;
end;
```

Quá trình duyệt theo thứ tự trước bắt đầu bằng lời gọi Visit(nút gốc).

Như cây ở Hình 24, nếu ta duyệt theo thứ tự trước thì các giá trị sẽ lần lượt được liệt kê theo thứ tự:

A B D H I E J C F K G L

6.4.2. Duyệt theo thứ tự giữa (inorder traversal)

Trong phép duyệt theo thứ tự giữa thì giá trị trong mỗi nút bất kỳ sẽ được liệt kê sau giá trị lưu ở nút con trái và được liệt kê trước giá trị lưu ở nút con phải của nút đó, có thể mô tả bằng thủ tục đệ quy sau:

```
procedure Visit(N); {Duyệt nhánh cây nhận N là nút gốc của nhánh đó}
begin
  if N ≠ nil then
    begin
      Visit(Nút con trái của N);
      <Output trường Info của nút N>
      Visit(Nút con phải của N);
    end;
end;
```

Quá trình duyệt theo thứ tự giữa cũng bắt đầu bằng lời gọi Visit(nút gốc).

Như cây ở Hình 24, nếu ta duyệt theo thứ tự giữa thì các giá trị sẽ lần lượt được liệt kê theo thứ tự:

H D I B E J A K F C G L

6.4.3. Duyệt theo thứ tự sau (postorder traversal)

Trong phép duyệt theo thứ tự sau thì giá trị trong mỗi nút bất kỳ sẽ được liệt kê sau giá trị lưu ở hai nút con của nút đó, có thể mô tả bằng thủ tục đệ quy sau:

```
procedure Visit(N); {Duyệt nhánh cây nhận N là nút gốc của nhánh đó}
begin
  if N ≠ nil then
    begin
      Visit(Nút con trái của N);
      Visit(Nút con phải của N);
      <Output trường Info của nút N>
    end;
end;
```

Quá trình duyệt theo thứ tự sau cũng bắt đầu bằng lời gọi Visit(nút gốc).

Cũng với cây ở Hình 24, nếu ta duyệt theo thứ tự sau thì các giá trị sẽ lần lượt được liệt kê theo thứ tự:

H I D J E B K F L G C A

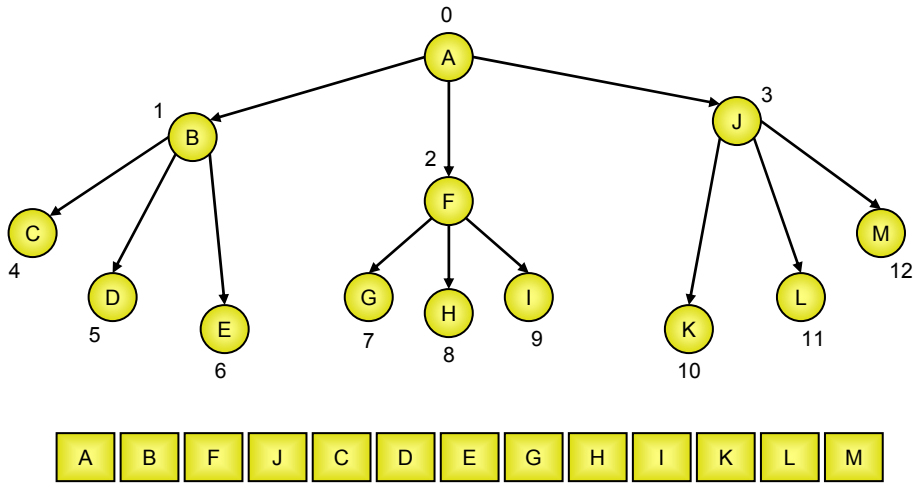
6.5. CÂY K_PHÂN

Cây K_phân là một dạng cấu trúc cây mà mỗi nút trên cây có tối đa K nút con (có tính đến thứ tự của các nút con).

6.5.1. Biểu diễn cây K_phân bằng mảng

Cũng tương tự như việc biểu diễn cây nhị phân, người ta có thể thêm vào cây K_phân một số nút giả để cho mỗi nút nhánh của cây K_phân đều có đúng K nút con, các nút con được xếp thứ tự từ nút con thứ nhất tới nút con thứ K, sau đó đánh số các nút trên cây K_phân bắt đầu từ 0 trở đi, bắt đầu từ mức 1, hết mức này đến mức khác và từ "trái qua phải" ở mỗi mức.

Theo cách đánh số này, nút con thứ j của nút i là: $K.i + j$. Nếu i không phải là nút gốc (I > 0) thì nút cha của nút i là nút $\left\lfloor \frac{i-1}{K} \right\rfloor$. Ta có thể dùng một mảng T đánh số từ 0 để lưu các giá trị trên các nút: Giá trị tại nút thứ i được lưu trữ ở phần tử T[i].



Hình 25: Đánh số các nút của cây 3_phân để biểu diễn bằng mảng

6.5.2. Biểu diễn cây K_phân bằng cấu trúc liên kết

Khi biểu diễn cây K_phân bằng cấu trúc liên kết, mỗi nút của cây là một bản ghi (record) gồm hai trường:

- Trường Info: Chứa giá trị lưu trong nút đó.
- Trường Links: Là một mảng gồm K phần tử, phần tử thứ i chứa liên kết (con trỏ) tới nút con thứ i, trong trường hợp không có nút con thứ i thì Links[i] được gán một giá trị đặc biệt.

Đối với cây K_ phân, ta cũng chỉ cần giữ lại nút gốc, bởi từ nút gốc, đi theo các hướng liên kết có thể đi tới mọi nút khác.

6.6. CÂY TỔNG QUÁT

Trong thực tế, có một số ứng dụng đòi hỏi một cấu trúc dữ liệu dạng cây nhưng không có ràng buộc gì về số con của một nút trên cây, ví dụ như cấu trúc thư mục trên đĩa hay hệ thống đề mục của một cuốn sách. Khi đó, ta phải tìm cách mô tả một cách khoa học cấu trúc dữ liệu dạng cây tổng quát. Cũng như trường hợp cây nhị phân, người ta thường biểu diễn cây tổng quát bằng hai cách: Lưu trữ kế tiếp bằng mảng và lưu trữ bằng cấu trúc liên kết.

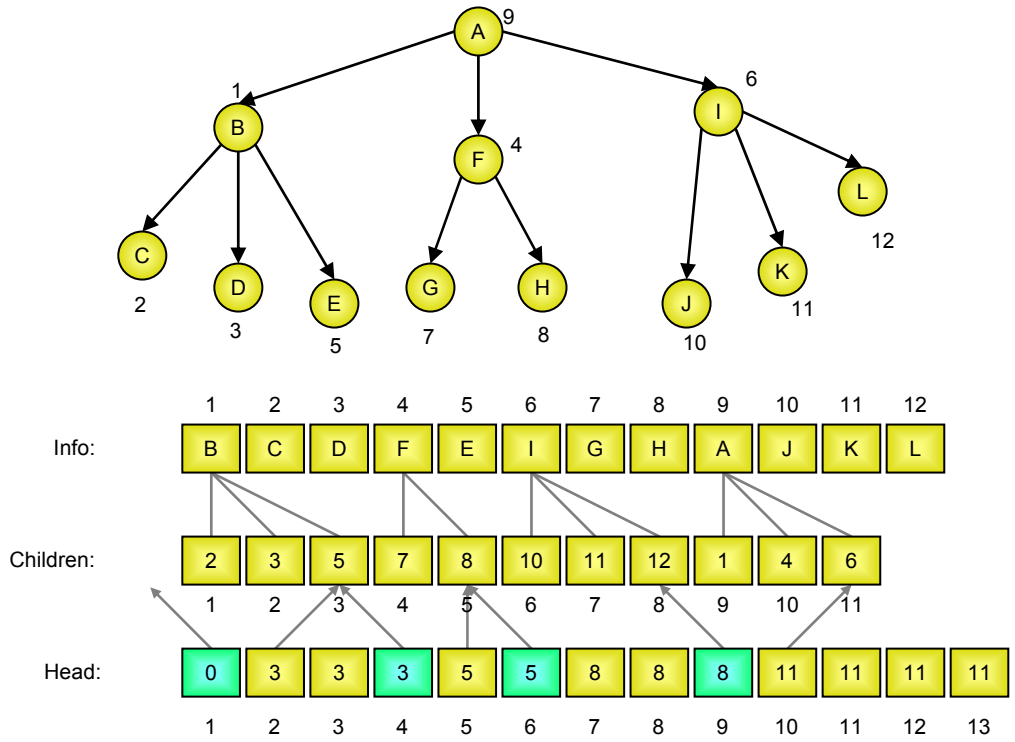
6.6.1. Biểu diễn cây tổng quát bằng mảng

Để lưu trữ cây tổng quát bằng mảng, giả sử cây có n nút, ta đánh số các nút trên cây bằng các số tự nhiên từ 1 tới n theo một thứ tự tuỳ ý. Cấu trúc dữ liệu để biểu diễn cây gồm có:

- Một mảng Info[1..n], trong đó Info[i] là giá trị lưu trong nút thứ i.
- Một mảng Children được chia làm n đoạn, đoạn thứ i gồm một dãy liên tiếp các phần tử là chỉ số các nút con của nút i. Nói cách khác, dùng mảng Children để liệt kê các nút con của nút 1, tiếp theo đến các nút con của nút 2, … theo đúng thứ tự đánh chỉ số. Mảng Children sẽ chứa tất cả chỉ số của mọi nút con trên cây (ngoại trừ nút gốc) nên nó sẽ gồm n - 1 phần tử. Lưu ý rằng khi chia mảng Children làm n đoạn thì sẽ có những đoạn rỗng (tương ứng với danh sách các nút con của một nút lá)

- Một mảng Head[1..n + 1], để đánh dấu vị trí cắt đoạn trong mảng Children: Head[i] là vị trí đứng liền trước đoạn thứ i, hay nói cách khác: Đoạn con của mảng Children tính từ chỉ số Head[i] + 1 đến Head[i] sẽ được dùng để chứa chỉ số các nút con của nút thứ i. Khi Head[i] = Head[i + 1] có nghĩa là đoạn thứ i rỗng. Quy ước: Head[n + 1] = n - 1.
- Một biến lưu chỉ số của nút gốc.

Ví dụ:

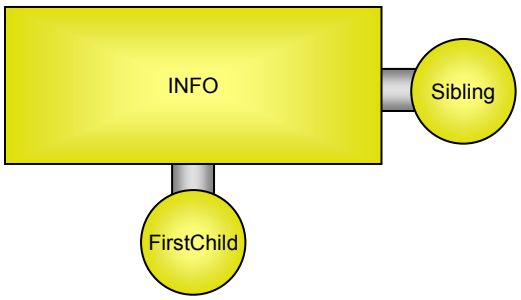


Hình 26: Biểu diễn cây tổng quát bằng mảng

6.6.2. Lưu trữ cây tổng quát bằng cấu trúc liên kết

Khi lưu trữ cây tổng quát bằng cấu trúc liên kết, mỗi nút là một bản ghi (record) gồm ba trường:

- Trường Info: Chứa giá trị lưu trong nút đó.
- Trường FirstChild: Chứa liên kết (con trỏ) tới nút con đầu tiên của nút đó (con cả), trong trường hợp là nút lá (không có nút con), trường này được gán một giá trị đặc biệt.
- Trường Sibling: Chứa liên kết (con trỏ) tới nút em kế cận bên phải (nút cùng cha với nút đang xét, khi sắp thứ tự các con thì nút đó đứng liền sau nút đang xét). Trong trường hợp không có nút em kế cận bên phải, trường này được gán một giá trị đặc biệt.



Hình 27: Cấu trúc nút của cây tổng quát

Dễ thấy được tính đúng đắn của phương pháp biểu diễn, bởi từ một nút N bất kỳ, ta có thể đi theo liên kết FirstChild để đến nút con cả, nút này chính là chốt của một danh sách nối đơn các nút con của nút N: từ nút con cả, đi theo liên kết Sibling, ta có thể duyệt tất cả các nút con của nút N.

Bài tập

Bài 1

Viết chương trình mô tả cây nhị phân dùng cấu trúc liên kết, mỗi nút chứa một số nguyên, và viết các thủ tục duyệt trước, giữa, sau.

Bài 2

Chứng minh rằng nếu cây nhị phân có x nút lá và y nút cấp 2 thì x = y + 1

Bài 3

Chứng minh rằng nếu ta biết dãy các nút được thăm của một cây nhị phân khi duyệt theo thứ tự trước và thứ tự giữa thì có thể dựng được cây nhị phân đó. Điều này con đúng nữa không đối với thứ tự trước và thứ tự sau? Với thứ tự giữa và thứ tự sau.

Bài 4

Viết các thủ tục duyệt trước, giữa, sau không đệ quy.