

Tổng quan về cây nhị phân

Cây nhị phân (Binary Tree) là một cấu trúc phi tuyến đầu tiên mà ta tiếp cận (Là một dạng đồ thị đặc biệt). Với tính chất phi tuyến nên việc xây dựng các thuật toán trên cây trở nên phức tạp, không tường minh tức là rất khó để lập trình tuyến tính (Iteration). Cách tiếp cận cấu trúc này là bằng khái niệm tổng quát hóa. Tổng quát hóa để định nghĩa đối tượng và cũng bằng tổng quát hóa để xử lý đối tượng.

Định nghĩa. Một cây T được gọi là cây nhị phân nếu thỏa:

- T rỗng.
- T chỉ có đúng một nút (node).
- Giả sử T_1, T_2 là hai cây nhị phân, bằng phép xây dựng kết hợp một nút r với nhánh trái của nó là cây T_1 , nhánh phải của nó là cây T_2 ta thu được cây T với nút gốc là r .

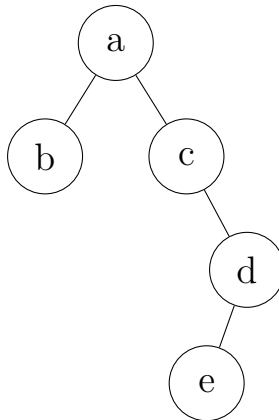
Xét ví dụ:

Cây T rỗng, con trỏ $r = NULL$

Cây chỉ có một nút trở bởi r .



Cây nút gốc trở bởi r với hai cây con trái và phải.



Như vậy, cấu trúc dữ liệu cho cây nhị phân được định nghĩa như sau:

```

typedef struct node {
    char data;          //du lieu luu o nut goc
    struct node*left;  //dia chi cua cay con trai
    struct node*right; //dia chi cua cay con phai
}node;
  
```

Mặc dù không có ứng dụng nhiều đối với cây nhị phân, tuy nhiên phép toán duyệt cây là cơ bản giúp người lập trình đầu tiên tiếp cận với cấu trúc dữ liệu đặc biệt này.

Bài toán: Cho một cây nhị phân, viết chương trình in các nút của cây theo ba thứ tự duyệt.

- Duyệt trước.
- Duyệt giữa.
- Duyệt sau.

Dữ liệu vào:

- Một dòng duy nhất chứa xâu S bao gồm các ký tự Alphabet và ký tự 0. Trong đó ký tự Alphabet là dữ liệu của nút, ký tự 0 sẽ được thay thế bằng ký tự NULL khi xử lý thuật toán tạo cây nhị phân.

Dữ liệu ra:

- In ra dòng thứ nhất là thứ tự duyệt trước, dòng thứ hai là thứ tự duyệt giữa và dòng cuối là thứ tự duyệt sau.

Ràng buộc:

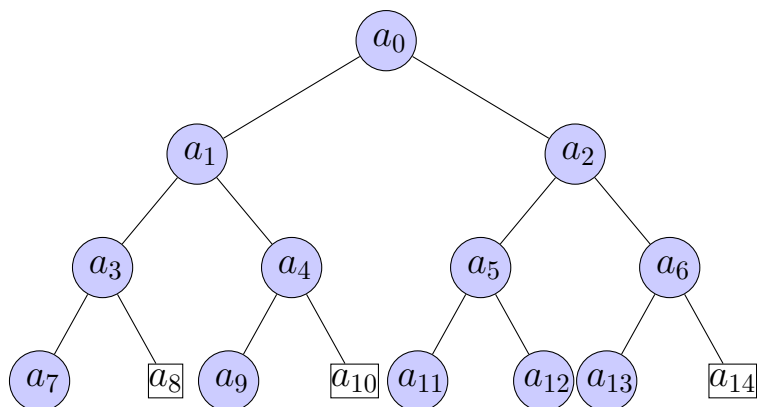
- Các giá trị: độ dài xâu S tối đa 100 ký tự.

Các ví dụ:

Dữ liệu vào:	Dữ liệu ra:
ABCDEF0G0000000000	A B D G E C F G D B E A F C G D E B F C A

Giải thích:

Do cây nhị phân định nghĩa đơn giản và chưa ràng buộc chặt nên hầu như phép chèn, xóa (2 phép toán cơ bản) lại không thể cài đặt được. Đối với bài toán này việc tạo ra cây nhị phân với dữ liệu vào như trên ta mô phỏng cấu trúc Heap cho cây nhị phân. Cho một dãy các phần tử a_0, a_1, \dots, a_{n-1} , ta tạo ra cây nhị phân theo quy tắc phân phối thể hiện như hình vẽ sau:



Với việc phân phối như trên ta sẽ tạo ra được cây như ý muốn, ở nhánh (trái, phải) nào khuyết thì phần tử a_i ở vị trí đó ta cho giá trị NULL (ví dụ nút a_8, a_{10}, a_{14}) và cuối cùng ta có cây nhị phân.

Để dễ dàng hơn trong giải bài toán với phép duyệt, ở đây cung cấp cho các bạn thuật toán chèn cây theo ý tưởng trên và bạn sẽ phải gọi lại hàm khi giải bài toán trên trên.

```
node* insert(char c[],int n) {
    node*tree=NULL;
    if (c[n]=='\0') c[n]='\0';//ky tu '\0' la gia tri NULL.
    if(c[n]!='\0') {
        tree=(node*)malloc(sizeof(node));
        tree->left=insert(c,2*n+1);
        tree->data=c[n];
        tree->right=insert(c,2*n+2);
    }
    return tree;
}
//call in main
node*tree=NULL;
tree=insert(c,0);
```