

Bài thực hành 8

1 Mục tiêu

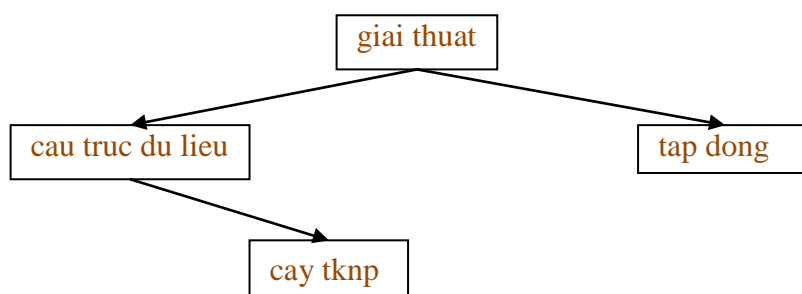
- Cài đặt cây tìm kiếm nhị phân

2 Bài tập lập trình bắt buộc

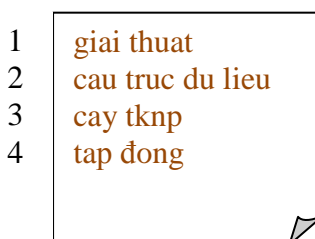
[bss.cpp]

Hãy cài đặt lớp BstStringSet biểu diễn một tập động các xâu (ví dụ tập từ vựng của tiếng Việt) dùng cây tìm kiếm nhị phân.

Ví dụ: tập động {“giải thuật”, “cấu trúc dữ liệu”, “cây tkn”, “tập động”} có thể cài bằng cây trong Hình 1.



Hình 1 Ví dụ tập động cài bằng cây tìm kiếm nhị phân



Hình 2 Tập vidu.txt chứa dữ liệu cho tập động ví dụ

Giao diện của lớp cần cung cấp các hàm sau đây

- 1) BstStringSet(): Hàm kiến tạo không tham số tạo tập rỗng.
- 2) BstStringSet(char * filename): Kiến tạo tập động từ dữ liệu của tệp văn bản có tên filename. Trong tệp này mỗi xâu được ghi trên một dòng. Ta cài hàm này bằng cách gọi tới hàm insert (lần lượt thêm các xâu vào cây tìm kiếm nhị phân).
- 3) ~BstStringSet(): Hủy tập động
- 4) bool find(const string & x) const: Xâu x có trong tập không?
- 5) void erase(const string & x): Nếu x có trong tập thì xóa x. Cây sau khi xóa x vẫn là cây tìm kiếm nhị phân.
- 6) void insert(const string & x): Nếu x không có trong tập thì chèn thêm x. Cây sau khi chèn x vẫn là cây tìm kiếm nhị phân.
- 7) string min() const: Trả về xâu đứng đầu tập động theo thứ tự từ điển.
- 8) string max() const: Trả về xâu đứng cuối tập động theo thứ tự từ điển.

- 9) `string predecessor(const string & x) const`: Trả về xâu đứng liền trước x trong tập động theo thứ tự từ điển.
- 10) `string successor(const string & x) const`: Trả về xâu đứng liền sau x trong tập động theo thứ tự từ điển.
- 11) `void printInorder() const`: In ra kết quả duyệt cây TKNP theo thứ tự trong.

Gợi ý:

- Mỗi đỉnh trong cây có thể có cấu trúc

```
struct Node{
    string data;
    Node * left;
    Node * right;
};
```
- Dữ liệu private của `BstStringSet` lưu địa chỉ đỉnh gốc của cây

```
Node * root;
```