



CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Bài Tập Lớn 2

THE LORD OF THE RINGS: THE TWO TOWERS

Phiên bản 1.1

1. Giới thiệu

Hành trình của Hiệp hội bảo vệ nhẫn rất gian nan và nguy hiểm khi luôn luôn bị thế lực bóng tối của chúa tể Sauron ngăn cản. Đã có người hy sinh, bị bắt trên suốt dọc hành trình. Frodo đã biến mất cùng với chiếc nhẫn. Aragorn (chính là hiệp sĩ Strider) và những người còn lại tiếp tục cuộc hành trình (có phần vô vọng). Khi đó họ đã gặp lại pháp sư Gandalf, người đã tái sinh mạnh mẽ hơn sau khi rơi xuống địa ngục trong cuộc chiến đấu ở phần trước với Balrog, một ác quỷ cổ xưa của lửa và bóng tối. Cùng đó, những người cây Ents cũng ngăn chặn Giáo chủ Saruman, người đã ngã về phe Chúa tể bóng tối và đang phá hủy một phần lớn khu rừng để làm chất đốt cho những lò rèn vũ khí của mình.

Aragorn, người lùn Gimli và vị tiên Legolas cùng với Gandalf hợp lực cùng các đội quân khác quyết tâm chống lại quân đội ngày càng qui mô của Saruman. Sau khi giải thoát vua Théoden của xứ sở Edoras khỏi sự khống chế của Saruman, cuộc chiến tại thành Helm's Deep bắt đầu với binh lực của phe sáng yếu hơn nhiều so với phe bóng tối. Tuy nhiên, vào lúc tưởng chừng như toàn bộ thành trì sụp đổ, Gandalf đã trở lại, sau khi rời đi tìm quân cứu viện trước đó. Và cùng với Gandalf là đội quân Éomer. Họ đã lật ngược thế cờ, đánh bại đội quân bóng tối. Thành lũy đã được bảo vệ, một bức tranh tươi sáng hơn đã dần hiện lên.

Từ thắng lợi này, cùng với sự hợp lực của những người cây Ents, họ quyết định tấn công Toà tháp đôi, thành lũy của Saruman, nơi Saruman tạo ra các đoàn quân và vũ khí hùng mạnh của mình. Đồng thời, họ còn muốn lấy lại quả cầu tiên tri, vật có thể giúp Chúa tể bóng tối theo dõi bất kỳ hoạt động nào của đội quân ánh sáng và khống chế các vị vua như Théoden. Một cuộc chiến công thành mới cam go hơn bắt đầu, quyết định thế trận của cuộc chiến với Chúa tể bóng tối.

Aragorn, bây giờ được biết là dòng dõi nhà vua xứ Gondor, phát ra lời kêu gọi mời anh tài tụ hội nơi khu rừng Fangorn, lãnh địa của người cây Ents, để chuẩn bị tấn công Toà tháp đôi. Vào thời này, chiến thuật để tấn công một thành trì rất đơn giản: sau những trận mưa tên, một mũi tấn công sẽ đột kích thẳng vào cổng chính thành trì, trong khi những chiến binh khác sẽ hỗ trợ từ hai cánh. Sau này, các nhà khoa học đã khám phá rằng đội hình mà các hiệp sĩ thời đó sử dụng chính là một dạng *cây nhị phân* (binary tree).

2. Yêu cầu

Trong bài tập lớn này, sinh viên sẽ được cung cấp một file chứa dữ liệu nhập, bao gồm các sự kiện xảy ra trong suốt quá trình tấn công Toà tháp đôi. Chương trình sẽ xuất ra đội hình các hiệp sĩ còn lại sau khi chiến thắng. Dữ liệu nhập mà sinh viên phải xử lý được biểu diễn dưới dạng *danh sách liên kết* (linked list), dữ liệu xuất sẽ biểu diễn dưới dạng *cây nhị phân*. Chi tiết cụ thể công việc sinh viên phải làm sẽ mô tả trong phần 4.

3. Dữ liệu nhập

Dữ liệu nhập của chương trình được chứa trong file mang tên `input.txt`. File này sẽ chứa danh sách các sự kiện xảy ra tại cuộc chiến. Mỗi sự kiện sẽ được mô tả bằng một giá trị số, gọi là *mã sự kiện*. Ý nghĩa tương ứng của từng sự kiện được mô tả trong [Bảng 1](#). Số sự kiện là không cố định, có thể thay đổi tùy theo test case. Một sự kiện có thể xảy ra nhiều lần. Sẽ có tối đa 1000 sự kiện xảy ra. Nếu số sự kiện nhiều, các sự kiện có thể trình bày thành nhiều dòng.

Bảng 1 – Các sự kiện xảy ra trong cuộc tấn công Toà tháp đôi.

Mã sự kiện	Ý nghĩa
-XYZL	Gặp quái vật phòng thủ lâu đài, quái vật này có <i>key</i> là XYZ và có level là L.
1XYZL	Một hiệp sĩ có <i>key</i> là XYZ và có level là L vừa đến Khu rừng Fangorn.
2XYZ	Quả cầu tiên tri được giải cứu bởi hiệp sĩ có <i>key</i> gần với XYZ nhất
3XYZ	Hiệp sĩ có <i>key</i> gần với XYZ nhất vừa gặp nữ vương Galadriel
5	Sự kiện Saruman bỏ chạy trốn khỏi Toà tháp đôi

4. Hiện thực chương trình

Sinh viên sẽ hiện thực một hàm *siege* có prototype như sau:

KnightTree siege(eventList* pEvent)*

Thông số *pEvent* là một con trỏ trỏ đến danh sách liên kết của các sự kiện được đọc từ file input, được định nghĩa như sau:

```
struct eventList {
    int nEventCode;
    eventList* pNext;
}
```

KnightTree là cấu trúc cây nhị phân mô tả đội hình chiến đấu của các hiệp sĩ khi tấn công Toà tháp đôi, có cấu trúc như sau:

```
struct KnightTree {
    int key;
    int level;
    int balance; //will be used in AVL only, and be ignored in other cases.
    KnightTree* pLeftChild;
    KnightTree* pRightChild;
}
```

Như vậy, mỗi hiệp sĩ sẽ được biểu diễn như một nút trên cây, thông tin về hiệp sĩ bao gồm *key* của hiệp sĩ khi nhập hội và *level* của hiệp sĩ. Giá trị của *key* nằm trong khoảng [0-999], giá trị của *level* nằm trong khoảng [0-9]. (Xem thêm phần 5 để biết thêm về *key*).

Ví dụ 1: Nếu có hai hiệp sĩ tham gia tấn công lâu đài, hiệp sĩ ở vị trí nút gốc có *key* là 111 và *level* là 3, hiệp sĩ thứ hai đứng ở vị trí nút con bên phải của hiệp sĩ đầu, có *key* là 222 và *level* là 4; thì thông tin về đội hình của hai hiệp sĩ này sẽ được biểu diễn là **(111_3 (N 222_4))**. N viết tắt cho NULL, tức nút gốc không có nút con bên trái

5. Xây dựng cây nhị phân kết quả

Cây nhị phân kết quả của hàm *siege* sẽ được xây dựng theo các nguyên tắc sau:

S1) Các hiệp sĩ khi đến nhập hội tại Khu rừng Fangorn đều mang theo mình một *số hiệu hiệp sĩ* gọi là *key* hoặc *số hiệu*. Đây chính là số hiệu mà Chúa nhẫn đã đặt cho các hiệp sĩ trước đó khi thế giới còn yên bình, trong một nỗ lực nhằm quản lý toàn bộ các hiệp sĩ. Căn cứ vào *số hiệu* của các hiệp sĩ, họ sẽ sắp xếp tạo thành một *cây nhị phân tìm kiếm* (binary search tree), trong đó hiệp sĩ đến đầu tiên sẽ đóng vai trò là nút gốc, các hiệp sĩ đến sau tùy theo giá trị *key* của mình mà tham gia vào đội hình chiến đấu.

Ví dụ 2: Với dữ liệu nhập là

17234 19343 12246 19566

Cây nhị phân kết quả sẽ là **(723_4 (224_6 934_3 (N 956_6)))**.

Kể từ khi có hiệp sĩ đầu tiên đến Khu rừng Fangorn, nghĩa là từ khi cây nhị phân hiện hành có nút đầu tiên, số nút trên cây nhị phân sẽ thay đổi tùy theo các sự kiện xảy ra, như được mô tả bên dưới. Tuy nhiên, nếu sau một sự kiện nào đó, **cây nhị phân không còn tồn tại nút nào thì hàm *siege* chấm dứt ngay lập tức và trả về kết quả là NULL** (xem thêm Ví dụ 8 bên dưới).

S2) Nếu khi đến Khu rừng Fangorn, số hiệu của hiệp sĩ trùng với số hiệu của một hiệp sĩ đã đến nhập hội trước đó, **số hiệu của hiệp sĩ sẽ được tăng dần từng đơn vị cho đến giá trị đầu tiên không trùng với bất kỳ số hiệu nào của các hiệp sĩ đã nhập hội.**

Ví dụ 3: Với dữ liệu nhập là

17234 17243 17268 17239

Sau khi ba hiệp sĩ đầu tiên (có số hiệu lần lượt là 723, 724, và 726) nhập hội, do số hiệu của hiệp sĩ thứ ba là 723 trùng với số hiệu của hiệp sĩ đầu nên số hiệu của hiệp sĩ thứ ba sẽ được tăng dần đến số 725. Cây nhị phân kết quả sẽ là **(723_4 (N 724_3 (N 726_8 (725_9 N))))**.



Trong lúc tăng dần số hiệu của hiệp sĩ để giải quyết đung độ, số hiệu của hiệp sĩ không được nhận các giá trị đặc biệt 777, 888, 999. (Ví dụ nếu số hiệu của hiệp sĩ đang là 776 thì số hiệu tăng dần tiếp theo sẽ là 778). Nếu số hiệu của hiệp sĩ đã tăng đến giá trị 998 mà vẫn không thể nhập hội, hiệp sĩ sẽ bị nghi ngờ là gián điệp của Chúa tể bóng tối và sẽ bị từ chối nhập hội.

S3) Khi gặp mã sự kiện có dạng $2XYZ$, Quả cầu tiên tri đã được giải cứu bởi hiệp sĩ có số hiệu ABC gần với XYZ nhất. Quả cầu tiên tri sẽ chỉ xuất hiện một lần trong toàn bộ các sự kiện.

Định nghĩa 1. Nút có số hiệu ABC được xem là gần với XYZ nhất trên toàn cây nhị phân nếu $|ABC - XYZ|$ là nhỏ nhất so với các nút khác. Nếu có hai nút có cùng giá trị gần nhất như vậy, sẽ ưu tiên cho nút có key nhỏ hơn.

Ví dụ 4: Với dữ liệu nhập là

17234 17243 17239 2724

Khi gặp Quả cầu tiên tri, giá trị của cây nhị phân hiện hành là $(723_4 (N 724_3 (N 725_9)))$; do đó hiệp sĩ cứu được Quả cầu tiên tri là hiệp sĩ có số hiệu **724**.

Ví dụ 5: Với dữ liệu nhập là

17234 17253 2724

Khi gặp Quả cầu tiên tri, giá trị của cây nhị phân hiện hành là $(723_4 (N 725_3))$; hiệp sĩ cứu được Quả cầu tiên tri là hiệp sĩ có số hiệu **723**.

Khi cứu được Quả cầu tiên tri, cuộc chiến vẫn tiếp tục để tiêu diệt toàn bộ Toà tháp đôi, một hang ổ của Chúa tể bóng tối. Với quyền năng đặc biệt của mình, Quả cầu tiên tri sẽ tăng level của hiệp sĩ cứu mình lên một đơn vị, tuy nhiên level của hiệp sĩ không được vượt quá 9.

Ví dụ 6: Với dữ liệu nhập là

17234 17253 2724 17246

Cây nhị phân kết quả sẽ là $(723_5 (N 725_3 (724_6 N)))$.

S4) Nếu các hiệp sĩ tấn công gặp một quái vật bảo vệ lâu đài có số hiệu là XYZ , hiệp sĩ có số hiệu ABC gần với XYZ nhất sẽ giao đấu với quái vật này. Nếu hiệp sĩ có level lớn hơn hoặc bằng level của quái vật, hiệp sĩ thắng, ngược lại hiệp sĩ sẽ bị thua. Nếu bị thua, hiệp sĩ sẽ bị loại khỏi cây nhị phân hiện hành. Nếu nút tương ứng với hiệp sĩ có hai nhánh con, việc xóa nút này ra khỏi cây sẽ áp dụng giải thuật in-order successor (http://en.wikipedia.org/wiki/Binary_search_tree). Theo đó, node tận cùng trái của cây con bên phải node cần xóa sẽ được đưa lên thay thế. Quái vật sẽ không xuất hiện nếu chưa có hiệp sĩ nào đến Khu rừng Fangorn.

Ví dụ 7: Với dữ liệu nhập là

17234 17223 17246 -7235



Trước khi gặp quái vật ở sự kiện thứ tư, cây nhị phân hiện hành sẽ là (723_4 (722_3 724_6)). Hiệp sĩ có số hiệu 723 sẽ giao đấu với quái vật và bị thua, cây nhị phân kết quả sẽ là (724_6 (722_3 N)).

Ví dụ 8: Với dữ liệu nhập là

17234 -1235 17223 17246 -7235

Trước khi gặp quái vật ở sự kiện thứ hai, cây nhị phân hiện hành sẽ là (723_4). Hiệp sĩ có số hiệu 723 sẽ giao đấu với quái vật và bị thua, cây nhị phân hiện hành lúc này rỗng (không còn nút nào). Hàm *siege* sẽ kết thúc và trả về kết quả NULL.

S5) Nếu hiệp sĩ nhập hội có số hiệu 777, hiệp sĩ này chính là Aragorn. Theo luật của khu rừng Fangorn, Aragorn luôn được đứng ở vị trí nút gốc của đội hình chiến đấu. Như vậy khi Aragorn đến Khu rừng Fangorn, đội hình chiến đấu của các hiệp sĩ sẽ được sắp xếp lại như sau: Các hiệp sĩ trên cây hiện hành sẽ được **sắp lại thành một danh sách theo thứ tự duyệt cây NLR**. Sau đó một cây mới được tạo ra bằng cách đưa Aragorn vào vị trí nút gốc, sau đó các hiệp sĩ trong danh sách sẽ được lần lượt đưa lại vào cây.

Ví dụ 9: Với dữ liệu nhập là

17234 17223 17246 17771 18234

Trước khi Aragorn xuất hiện ở sự kiện thứ tư, cây nhị phân hiện hành sẽ là (723_5 (722_3 724_6)), như vậy khi duyệt theo thứ tự NLR sẽ là [723_5, 722_3, 724_6]. Sau khi tạo cây mới với Aragorn ở nút gốc, các hiệp sĩ trong danh sách lần lượt được thêm vào cây sẽ tạo thành cây mới (777_1 (723_5 (722_3 724_6) N)). Hiệp sĩ tiếp theo được thêm vào cây sẽ tạo thành cây kết quả sau cùng (777_1 (723_4 (722_3 724_6)) 823_4)).

Khi Aragorn đứng ở nút gốc, khi các hiệp sĩ công thành nếu gặp quái vật có cùng level với Aragorn (trừ ác quỷ Balrog), Aragorn sẽ dùng thanh gươm Narsil (của vua của loài người Elendil) hạ ngay quái vật này mà không cần giao đấu. *Lưu ý*, Aragorn chỉ dùng thanh gươm này nếu đang ở nút gốc (đang chỉ huy). Aragorn sẽ chiến đấu như các chiến binh bình thường khác nếu không ở nút gốc (không chỉ huy).

Ví dụ 10: Với dữ liệu nhập là

17234 17223 17246 17771 -1231 -7762

Tương tự Ví dụ 7, khi Aragorn nhập hội, cây nhị phân hiện hành sẽ là (777_1 (723_5 (722_3 724_6) N)). Sau đó quái vật tiếp theo (có số hiệu 123) có level là 1 (cùng level với Aragorn) nên sẽ bị Aragorn tiêu diệt ngay. Sau đó Aragorn giao đấu với quái vật có số hiệu 776 và bị thua nên sẽ bị loại ra khỏi cây. Cây nhị phân kết quả sẽ là (723_4 (722_3 724_6)).

S6) Nếu quái vật giữ thành có số hiệu là 777, đó chính là Đại quái vật Lurtz Mặt ngựa. Mặt ngựa sẽ tham chiến như một chiến binh bình thường. Tuy nhiên nếu số nút lá của cây nhị phân hiện hành bằng đúng với số level của Mặt ngựa, Mặt ngựa sẽ thực hiện tuyệt chiêu “Ngựa Hí” hất văng tất cả các hiệp sĩ ở *độ sâu* (trên cây) lớn hơn hoặc bằng số level của Mặt ngựa ra khỏi cây.

Định nghĩa 2: Độ sâu (depth) của một nút sẽ bằng khoảng cách từ nút đó đến nút gốc cộng thêm 1. Như vậy, độ sâu của nút gốc sẽ là 1

Ví dụ 11: Với dữ liệu nhập là

17779 -7771 2345 12345

Khi gặp Mặt ngựa, cây nhị phân lúc này là **(777_9)**. Do trên cây đang có đúng một nút lá, bằng với level của Mặt ngựa, Mặt ngựa sẽ xuất tuyệt chiêu Ngựa hí hất văng tất cả các nút có độ sâu lớn hơn hoặc bằng 1 ra khỏi cây. Do đó cây nhị phân lúc này trở nên rỗng, hàm *siege* kết thúc và trả về kết quả là NULL.

Ví dụ 12: Với dữ liệu nhập là

17771 -7771

Khi gặp Mặt ngựa, do level của Mặt ngựa trùng với level của Aragorn, Mặt ngựa chưa kịp xuất tuyệt chiêu đã bị triệt hạ, cây nhị phân kết quả sẽ là **(777_1)**.

S7) Nếu vệ sĩ giữ thành có số hiệu là 888, đó chính là Saruman. Saruman sẽ chiến đấu với các hiệp sĩ cho đến khi hoặc hắn bị đánh bại, hoặc đã giao đấu với tất cả các hiệp sĩ có trong cây. Nghĩa là, sau khi Saruman đánh thắng một trận, hắn lại tiếp tục tìm hiệp sĩ có số hiệu gần với hắn nhất trên cây nhị phân còn lại để tiếp tục trận đánh, quá trình này sẽ lặp lại liên tục cho đến khi hắn bị đánh bại.

Ví dụ 13: Với dữ liệu nhập là

18211 11239 -8883

Trước khi gặp Saruman ở sự kiện thứ 3, cây nhị phân hiện hành là **(821_1 (123_9 N))**. Khi gặp Saruman, hiệp sĩ có số hiệu 821 gần với số hiệu của Saruman nhất nên sẽ ra giao đấu với hắn trước và bị thua, cây nhị phân còn lại sẽ là **(123_9)**. Hiệp sĩ có số hiệu 123 sẽ ra giao đấu với Saruman và chiến thắng. Cây nhị phân kết quả sẽ là **(123_9)**.

S8) Nếu hiệp sĩ đến Khu rừng Fangorn có số hiệu là 000, hiệp sĩ đó chính là người lùn Gimli. Gimli sẽ nhập hội nếu cây nhị phân hiện hành không phải là một cây đầy đủ (*complete tree*).

Định nghĩa 3: Một cây nhị phân được gọi là đầy đủ (complete) nếu số nút N và chiều cao H của cây thỏa điều kiện $N = 2^H - 1$.

Ví dụ 14: Với dữ liệu nhập là

18211 10004

Khi Gimli đến nhập hội, cây nhị phân hiện hành là **(822_1)**. Đây là một cây đầy đủ nên Gimli không nhập hội, cây kết quả trả về là **(822_1)**.

Ví dụ 15: Với dữ liệu nhập là

18211 12345 10004

Khi Gimli đến nhập hội, cây nhị phân chưa hoàn toàn nên Gimli sẽ nhập hội, cây kết quả trả về là **(822_1 (234_5 (000_4 N) N))**.

S9) Nếu hiệp sĩ đến Khu rừng Fangorn có số hiệu là 999, hiệp sĩ đó chính là Pháp sư Gandalf. Ngay khi vừa nhập hội, Gandalf đã chứng tỏ mình có kiến thức quân sự kiệt xuất nên được nhất trí bầu làm chỉ huy quân tấn công. Khi nắm quyền chỉ huy, Gandalf sẽ yêu cầu các hiệp sĩ tấn công xếp thành đội hình cây cân bằng (cây AVL) để sức tấn công được trải đều trên mọi mặt trận. Khi Gandalf nhập hội, Các hiệp sĩ trên cây hiện hành sẽ được **sắp lại thành một danh sách theo thứ tự duyệt cây RNL**. Sau đó một cây mới được tạo ra bằng cách đưa Gandalf vào vị trí nút gốc, tiếp đó các hiệp sĩ trong danh sách sẽ được lần lượt đưa lại vào cây.

Do chỉ có Gandalf nắm được bí quyết vận hành cây AVL nên nếu ông bị đánh bại và bị loại khỏi cây, các hiệp sĩ sẽ tiếp tục vận hành đội hình cây theo các quy luật của cây nhị phân tìm kiếm như cũ. (Việc loại nút tương ứng với Gandalf ra khỏi cây cân bằng cũng sẽ tuân theo các quy tắc vận hành cây nhị phân tìm kiếm, không cần cân bằng lại cây sau khi xoá nút). Vốn kính phục Gandalf, nên nếu Gandalf nhập hội, Aragorn sẽ **chỉ** trở thành chỉ huy nếu nhập hội sau hoặc vô tình trở thành nút gốc một cách ngẫu nhiên (do quá trình sắp xếp hoặc loại bỏ nút gốc trên cây nhị phân mà có).

Ví dụ 16: Với dữ liệu nhập là

11231 12345 19994 -9985 10565

Khi Gandalf đến Khu rừng Fangorn, cây nhị phân hiện hành đang là **(123_1 (N 234_5))**, như vậy khi duyệt theo thứ tự RNL sẽ là **[234_5, 123_1]**. Sau đó Gandalf được đưa vào nút gốc của cây mới và lần lượt thêm các nút của danh sách RNL vào cây theo quy tắc vận hành cây AVL, ta sẽ được cây mới là **(234_5 (123_1 999_4))**. Sau đó Gandalf bại trận và bị loại khỏi cây, cây nhị phân hiện hành sẽ trở thành **(234_5 (123_1 N))**. Sau đó khi hiệp sĩ có số hiệu 056 nhập hội, cây nhị phân kết quả sẽ là **(234_5 (123_1 (056_5 N) N))**.

Ví dụ 17: Với dữ liệu nhập là

11231 12345 19994 17775 -9985 10565

Sau khi Gandalf đến Khu rừng Fangorn, cây AVL hiện hành là **(234_5 (123_1 999_4))**. Sau đó Aragorn nhập hội (sau Gandalf) nên cây được tổ chức thành **(777_5(234_5 (123_1 N) 9994))** Cuối cùng khi hiệp sĩ có số hiệu 056 nhập hội, cây nhị phân kết quả sẽ là **777_5(234_5 (123_1 (056_5 N) N) 999_4)**.

Ví dụ 18: Với dữ liệu nhập là

17771 18345 19994

Khi Gandalf đến Khu rừng Fangorn, cây nhị phân hiện hành đang là **(777_1 (N 834_5))**, như vậy khi duyệt theo thứ tự RNL sẽ là **[834_5, 777_1]**. Sau đó Gandalf được đưa vào nút gốc của cây mới và lần

lượt thêm các nút của danh sách RNL vào cây theo quy tắc vận hành cây AVL, ta sẽ được cây kết quả là (834_5 (777_1 999_4)).

S10) Nếu hiệp sĩ đến Khu rừng Fangorn có số hiệu là 888, hiệp sĩ đó chính là vị tiên Legolas. Vốn có cá tính táo bạo, vị tiên Legolas sẽ tổ chức lại các hiệp sĩ theo đội hình AVL-BST theo cách thức sau: Các chiến binh trên cây sẽ tạo thành danh sách theo thứ tự LNR, danh sách kết quả này sẽ có thứ tự (tăng dần). Sau đó vị tiên Legolas sẽ nhập vào danh sách này sau cho nó vẫn có thứ tự. Danh sách này sẽ được chuyển thành một cây AVL-BST theo cách sau: lấy nút giữa danh sách tạo thành nút gốc, gọi đệ qui để lấy phần đầu danh sách (trước phần tử giữa) tạo thành nhánh bên trái và gọi đệ qui để lấy phần sau danh sách (sau phần tử giữa) tạo thành nhánh bên phải của nút gốc.

Chú ý: mặc dù tạo ra cây AVL-BST, nhưng cây này được vận hành như một cây BST bình thường. Nghĩa là không cần cân bằng cây khi thêm vào và loại bỏ phần tử ra khỏi cây.

Ví dụ 19: Với dữ liệu nhập là

18901 11345 18884 -1338 10690

Khi vị tiên Legolas đến Khu rừng Fangorn, cây nhị phân hiện hành đang là (890_1 (134_5 N)), như vậy khi duyệt theo thứ tự LNR và thêm Legolas vào danh sách ta sẽ có [134_5, 888_4, 890_1]. Khi tạo cây AVL-BST từ danh sách này, ta sẽ có (888_4 (134_5 890_1)). Chú ý là cây này được vận hành theo cách BST bình thường. Khi gặp quái vật có số hiệu 133, hiệp sĩ 134 sẽ giao đấu và bị thua. Áp dụng giải thuật loại một nút ra khỏi BST bình thường, ta được cây nhị phân mới (888_4 (N 890_1)). Khi hiệp sĩ có số hiệu 10690 nhập hội, cây nhị phân sau cùng sẽ là (888_4 (0690 890_1)).

S11) Trong 3 người Gandalf, Aragorn và Legolas, ai là người đến khu rừng sau sẽ có quyền tổ chức lại cây bất kể trước đó cây đang được tổ chức theo hình thức nào.

S12) Nếu một hiệp sĩ gặp nữ vương Galadriel, nàng sẽ nâng level của hiệp sĩ lên 9.

S13) (Bonus) Trước sự tấn công mạnh mẽ của các hiệp sĩ, Saruman hiểu rằng Toà tháp đôi sớm muộn cũng sẽ thất thủ. Hắn bèn liều bắt cóc nữ vương Galadriel, người tình trong mộng của mình, và mở đường máu thoát khỏi trùng vây. Việc Saruman có thoát được vòng vây hay không phụ thuộc vào vận may của hắn: Nếu cây hiện tại gồm các level gộp lại chính là danh sách EC của Saruman thì hắn thoát được. Trong trường hợp này chương trình sẽ kết thúc và trả về cây nhị phân kết quả với level của các hiệp sĩ sẽ được gán bằng 0. (Xem phần 6 để biết thêm chi tiết về danh sách EC của Saruman). Ngược lại, chương trình cũng sẽ kết thúc và trả về cây nhị phân hiện tại.

Ví dụ 20: Nếu danh sách EC của Saruman là '126' và cây nhị phân sau cùng là (423_1 (012_6 456_2)) thì Saruman có cơ may thoát được khỏi vòng vây. Kết quả trả về là (423_0 (012_0 456_0)).

6. Cách dịch và thực thi chương trình

Sinh viên download file *assignment2.zip* từ trang Web của môn học. Khi giải nén file này, sẽ có được các file sau:

input.txt	Một file input ví dụ.
main.cpp	Chương trình chính
ttowers.cpp	Chương trình hiện thực bởi sinh viên
defs.h	File định nghĩa các cấu trúc và hàm dùng chung
Assignmen2.pdf	File mô tả nội dung bài tập lớn

File *input.txt* là một file nhập mẫu như được mô tả ở phần 3. File *main.cpp* là chương trình khởi tạo, bao gồm các hàm viết sẵn như sau:

- *main()*: chương trình chính sẽ thực thi
- *readFile()*: hàm đọc file input
- *display()* : hàm xuất dữ liệu ra màn hình.

Danh sách EC của Saruman sẽ được biểu diễn qua biến toàn cục *pSarumanList* khai báo trong file *defs.h*.

Lưu ý rằng **sinh viên không được phép thay đổi file *main.cpp* và *defs.h* khi hiện thực chương trình cũng như không được include bất kỳ thư viện nào khác (tất cả các thư viện cần thiết đều đã được include trong file *defs.h*)**. Ngoài ra, các hàm do sinh viên viết không được xuất bất kỳ dữ liệu nào ra màn hình khi thực thi.

Để dịch và thực thi chương trình, sinh viên chứa cả 3 files *main.cpp*, *ttowers.cpp* và *defs.h* trong cùng một thư mục; sau đó chỉ cần dịch và thực thi **duy nhất** file *main.cpp*. Mọi công việc cần phải làm sẽ được hiện thực trong file *ttowers.cpp*, tuy nhiên không cần dịch và thực thi file này.

Ví dụ 14: Để dịch và thực thi chương trình trên môi trường linux, có thể sử dụng lệnh sau:

```
$ make
```

```
$ ./assignment input.txt
```

7. Nộp bài

Khi nộp bài, sinh viên sử dụng account đã được cấp phát trên hệ thống BKEL và BKEL để nộp bài qua mạng. Sinh viên zip file *ttowers.cpp* cùng tối thiểu 30 testcase vào một file zip với tên *MSSV.zip*. **Tất cả các file sourcecode khác file *ttowers.cpp* sẽ bị tự động xóa khi chấm bài.**

File được nộp phải là file chương trình gốc, không có thêm cấp thư mục nào trong quá trình zip.

8. Thời hạn nộp bài

Thời hạn chót để nộp được thông báo cụ thể trên trang chính thức của môn học BKel. Sinh viên phải dùng account trên hệ thống Bkel và BKCSE để nộp bài. KHÔNG nhận bài được gửi qua mail hoặc bất kỳ hình thức nào khác. Bài nộp trễ sẽ KHÔNG được nhận.

9. Xử lý gian lận

Bài tập lớn phải được sinh viên TỰ LÀM. Sinh viên sẽ bị coi là gian lận nếu:

- Có sự giống nhau bất thường giữa mã nguồn của các bài nộp. Trong trường hợp này, TẤT CẢ các bài nộp đều bị coi là gian lận. *Do vậy sinh viên phải bảo vệ mã nguồn bài tập lớn của mình.*
- Sinh viên không hiểu mã nguồn do chính mình viết, trừ những phần mã được cung cấp sẵn trong chương trình khởi tạo. *Sinh viên có thể tham khảo từ bất kỳ nguồn tài liệu nào, tuy nhiên phải đảm bảo rằng mình hiểu rõ ý nghĩa của tất cả những dòng lệnh mà mình viết. Trong trường hợp không hiểu rõ mã nguồn của nơi mình tham khảo, sinh viên được đặc biệt cảnh báo là **KHÔNG ĐƯỢC** sử dụng mã nguồn này; thay vào đó nên sử dụng những gì đã được học để viết chương trình.*

Trong trường hợp bị kết luận là gian lận, sinh viên sẽ bị điểm 0 cho toàn bộ môn học (không chỉ bài tập lớn).

KHÔNG CHẤP NHẬN BẤT KỲ GIẢI THÍCH NÀO - KHÔNG CÓ BẤT KỲ NGOẠI LỆ NÀO!

Sau mỗi bài tập lớn được nộp, sẽ có một số sinh viên được gọi phỏng vấn ngẫu nhiên để chứng minh rằng bài tập lớn vừa được nộp là do chính mình làm.