



Đại Học Quốc Gia TP.HCM
Trường Đại học Bách Khoa
Khoa KH&KT Máy Tính

Vietnam National University – HCMC
Hochiminh University of Technology
Faculty of Computer Science and Engineering

CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

ASSIGNMENT 01

NARUTO VÀ BÀI KIỂM TRA ĐẦU TIÊN



1 Giới thiệu

Ở một thế giới khác, ninja là người có những năng lực đặc biệt. Họ sống tập trung thành từng ngôi làng với người đứng đầu gọi là Kage. Những đứa trẻ lớn lên trong những ngôi làng này đều được học những kỹ năng để trở thành một ninja thực thụ. Câu chuyện kể về cuộc đời của một cậu bé ninja của làng lá vô cùng đặc biệt: *Uzumaki Naruto*. Với ước mơ trở thành *Hokage*, và dường như đây là một số mệnh, con đường tương lai của cậu bé sẽ gặp vô vàn gian truân, trắc trở đã được báo trước từ khi có một thứ năng lượng khủng khiếp đang được chế ngự bên trong cậu bé này.

Và hôm nay, là ngày kiểm tra tốt nghiệp lớp ninja tập sự của những đứa trẻ trạc độ tuổi 12. Cuộc thi là một nhiệm vụ việc tìm kiếm một món đồ được chỉ định sẵn cho mỗi ninja và trên đường tìm kiếm sẽ gặp nhiều thử thách. Cuộc thi được tổ chức trong khu rừng bao quanh ngôi làng, vì thế để đảm bảo tính công bằng và an toàn, mỗi thí sinh được niệm một *nhãn thuật định vị*. Cứ sau mỗi chu kỳ thời gian nhất định, vị trí của các thí sinh sẽ được truyền về cho trung tâm dữ liệu của ban tổ chức.

2 Yêu cầu

Trong bài tập lớn này, sinh viên sẽ được cung cấp 2 tập tin chứa dữ liệu nhập: **data.csv** và **events.txt**. Chi tiết mô tả dữ liệu nhập, xuất và các công việc sinh viên phải làm được nêu trong các phần sau.

3 Dữ liệu

3.1 Dữ liệu nhập

- **data.csv**: chứa các thông tin về vị trí các ninja. Sinh viên chỉ cần sử dụng 4 thông số sau:
 - *REPORT_TIME* (timestamp): thời điểm gửi dữ liệu vị trí của ninja về trung tâm thông tin. Tất cả những thời điểm này đều trong cùng một ngày.
 - *NINJA_TAG*: số hiệu của ninja.
 - *LONGITUDE* (long): kinh độ.
 - *LATITUDE* (lat): vĩ độ.
- **events.txt**: chứa các yêu cầu xử lý dữ liệu. Các yêu cầu này được mô tả trong phần sau.

Như vậy nội dung tập tin dữ liệu nhập **events.txt** sẽ mô tả các yêu cầu theo thứ tự từ trái sang phải. Số lượng các yêu cầu là không cố định, có thể thay đổi tùy theo test case. Một yêu cầu có thể xuất hiện nhiều lần trong tập tin dữ liệu nhập. Nếu số yêu cầu nhiều, các yêu cầu có thể trình bày thành nhiều dòng. Mỗi yêu cầu được mô tả sẽ cách nhau bởi một dấu khoảng trắng, kết thúc các yêu cầu bằng dấu chấm phẩy. Nếu mã yêu cầu không tồn tại trong Bảng yêu cầu thì bỏ qua yêu cầu đó và đi tiếp đến yêu cầu tiếp theo.

3.2 Dữ liệu xuất

Dữ liệu xuất theo cú pháp sau: **<mã yêu cầu>: <kết quả>**

<kết quả> được in ra theo quy tắc sau:

- Số nguyên: không cần định dạng.
- Số thực: độ chính xác 12 chữ số.
- Danh sách: mỗi phần tử cách nhau bởi một dấu khoảng trắng và không có dấu khoảng trắng lúc kết thúc danh sách.

4 Quy ước và yêu cầu

Quy ước:

- Ninja được xem là đang dừng lại khi khoảng cách về vị trí giữa 2 thời điểm gửi tín hiệu tọa độ hiện tại và trước đó có độ lệch từ 5 mét trở xuống.

Yêu cầu:

- Sinh viên tự viết hàm đọc dữ liệu từ hai tập tin dữ liệu nhập data.csv và events.txt.
- Bảng các yêu cầu truy xuất dữ liệu nhập như sau:
 - (a) Một ninja cấp cao trang bị số hiệu XYZT sẽ tấn công một ninja đang tham gia cuộc thi có số hiệu gần nhất và nhỏ hơn XYZT. Ninja này sẽ bị loại khỏi cuộc thi. **Khi gặp yêu cầu này, in ra màn hình số hiệu của ninja bị loại.** Nếu không có ninja nào thỏa yêu cầu thì in ra **-1**.
 - (b) Naruto thường được cho là gốc gác vì thế để có thể hoàn thành nhiệm vụ của mình, anh ta phải mất khá nhiều thời gian để suy nghĩ. **Khi gặp yêu cầu này, in ra màn hình số hiệu của ninja có tổng thời gian dừng lại lớn nhất.**
 - (c) Mã yêu cầu này chứa một con số có 16 chữ số, cung cấp tọa độ địa lý (x, y) của hai điểm A và B, trong đó x là kinh độ và y là vĩ độ. Tọa độ 2 điểm A, B được xác định như sau:

Bảng 1: Các yêu cầu xử lý dữ liệu.

MÃ YÊU CẦU	Ý NGHĨA	GHI CHÚ
0	Danh sách các mã yêu cầu trong tập tin events.txt	In ra màn hình
1	Số hiệu ninja được lưu trữ đầu tiên.	In ra màn hình.
2	Số hiệu ninja được lưu trữ cuối cùng.	In ra màn hình.
3	Số lượng ninja đang tham gia cuộc thi.	In ra màn hình.
4	Số hiệu lớn nhất của ninja tham gia cuộc thi.	In ra màn hình.
5ABCD	Thời điểm lưu trữ khi ninja có số hiệu ABCD bắt đầu chuyển động lần đầu tiên trong ngày.	In ra màn hình. (Nếu không có ninja số hiệu ABCD là in ra số -1)
6ABCD	Thời điểm lưu trữ khi ninja có số hiệu ABCD bắt đầu dừng lại lần cuối cùng trong ngày.	In ra màn hình. (Nếu không có ninja số hiệu ABCD là in ra số -1)
7ABCD	Tổng số lần ninja có số hiệu ABCD dừng lại trong ngày.	In ra màn hình. (Nếu không có ninja số hiệu ABCD là in ra số -1)
8ABCD	Tổng quãng đường di chuyển trong ngày của ninja có số hiệu ABCD	In ra màn hình. (Nếu không có ninja số hiệu ABCD là in ra số -1)
9	Số hiệu của ninja có tổng quãng đường di chuyển trong ngày dài nhất.	In ra màn hình (Nếu có nhiều hơn 2 ninja thoả yêu cầu thì chọn ninja được lưu trữ đầu tiên)
10	Số hiệu của ninja có tổng thời gian di chuyển trong ngày nhiều nhất.	In ra màn hình. (Nếu có nhiều hơn 2 ninja thoả yêu cầu thì chọn ninja được lưu trữ đầu tiên)
11XYZT	Ninja giả mạo XYZT.	(a)
12	Phát hiện Naruto.	(b)
13ABCDmnpqEFGHuvs	Ninja bị lọt hố.	(c)
14	Phát hiện ninja bị lạc.	(d)

- Phần nguyên của x và y của 2 điểm A, B là phần nguyên của kinh độ và vĩ độ của ninja được lưu trữ (tọa các ninja được lưu trữ đều có cùng phần nguyên này).
- 4 chữ số **ABCD** là **phần thập phân của kinh độ** của điểm A (x_A).
- 4 chữ số **mnpq** là **phần thập phân của vĩ độ** của điểm A (y_A).
- 4 chữ số **EFGH** là **phần thập phân của kinh độ** của điểm B (x_B).
- 4 chữ số **uvrs** là **phần thập phân của vĩ độ** của điểm B (y_B).

Hai điểm A, B là hai đỉnh đối nhau của một hình vuông, gọi hình vuông này là **vùng cạm bẫy**. Ninja nào có đường đi rơi vào vùng cạm bẫy (kể cả viền hình vuông) sẽ bị lọt xuống hố. Kết thúc cuộc thi, ban tổ chức sẽ giải cứu các ninja trong **vùng cạm bẫy** này theo thứ tự ninja rơi vào vùng cạm bẫy sau sẽ được cứu ra trước. **Khi gặp yêu cầu này, in ra màn hình danh sách số hiệu các ninja lần lượt được cứu.** Nếu không có ninja nào thì in ra **-1**.

- (d) Khi một ninja bắt đầu di chuyển, nếu quãng đường của ninja này di chuyển tạo thành một đường khép kín thì xem như ninja này đang bị lạc. **Khi gặp yêu cầu này, in ra màn hình danh sách số hiệu các ninja bị lạc đường theo thứ tự lưu trữ.** Nếu không có ninja nào thì in ra **-1**. Lưu ý: khi kết thúc cuộc thi, ninja sẽ trở về nơi mà mình xuất phát, vì thế không tính quãng đường từ khi bắt đầu xuất phát cho đến khi kết thúc cuộc thi.

5 Hiện thực chương trình

Sinh viên được cung cấp các file sau:

- **main.cpp**: mã nguồn chính của chương trình
- **listLib.h**: file chứa định nghĩa thư viện danh sách liên kết
- **dbLib.h**: file header chứa prototype cần thiết để quản lý database
- **dbLib.cpp**: mã nguồn hiện thực các chức năng quản lý database
- **eventLib.h**: file header chứa các prototype cần thiết để quản lý các sự kiện
- **eventLib.cpp**: mã nguồn hiện thực các hàm xử lý sự kiện

Sinh viên được cho các hàm sau:

- **distanceEarth**: tính khoảng cách giữa 2 điểm trên mặt đất với tọa độ (latitude, longitude) cho trước.
- **strPrintTime**: in thời gian ra một chuỗi theo định dạng yêu cầu.

Sinh viên được yêu cầu hiện thực các hàm sau:

- **loadEvents**: đọc thông tin các sự kiện.
- **loadA01DB**: đọc file và lấy dữ liệu ninja vào danh sách.

Đối với file thư viện **listLib.h**, sinh viên được cung cấp sẵn định nghĩa cơ bản. Ngoài ra, để build được chương trình hoàn chỉnh, sinh viên cần hoàn thiện các hàm còn lại trong danh sách.

processData.cpp: sinh viên hiện thực việc xử lý các yêu cầu thông qua hàm **processEvent**. Không được thay đổi prototype của hàm này.

Sinh viên được phép tùy biến, chỉnh sửa **listLib.h** và **processData.cpp**.

Sinh viên không được sử dụng các thư viện nào khác ngoài các thư viện đã được dùng trong code mẫu.

5.1 Build

Sinh viên thực hiện build bằng lệnh **make** từ command line trên Linux và chạy file **a01**. Cú pháp trên linux như sau:

```
1 ./a01 event.txt data.csv
```

trong đó **event.txt** là file chứa danh sách các sự kiện, cách nhau bởi khoảng trắng hoặc ký tự xuống dòng. **data.csv** là tập tin chứa dữ liệu của ninja, sinh viên có thể thử nghiệm với các tập tin dữ liệu khác nhau.

Đối với các bạn dùng VisualStudio (VS) trên Windows, các bạn có thể tạo một project và tự thêm các file mã nguồn vào. Nếu không bạn có thể sử dụng hệ thống CMake để tự sinh project VS cho mình.

6 Quy định

6.1 Nộp bài

Sinh viên sẽ nộp bài qua hệ thống và sẽ được hướng dẫn cụ thể qua thông báo trên Sakai. Thời hạn nộp bài cũng sẽ được công bố trên Sakai khi bắt đầu mở hệ thống nhận bài. Sinh viên nộp một file nén duy nhất chứa tất cả các file mã nguồn.

Lưu ý các file **main.cpp**, **dbLib.h**, **eventLib.h** sẽ được ghi đè lúc chấm.

Sinh viên phải kiểm tra chương trình của mình trên Linux trước khi nộp.

6.2 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. Các bài làm của các sinh viên ở các học kỳ trước cũng sẽ được dùng để kiểm tra gian lận.
- 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 có NGOẠI LỆ nào được chấp nhận.**

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.