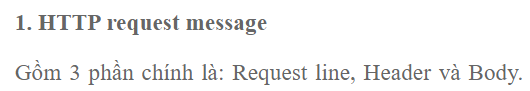
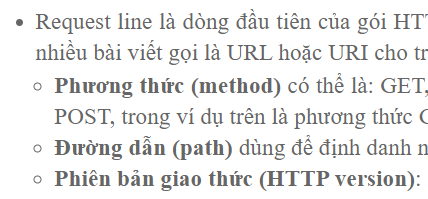
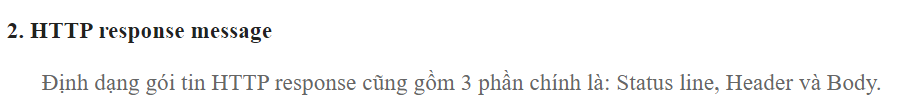
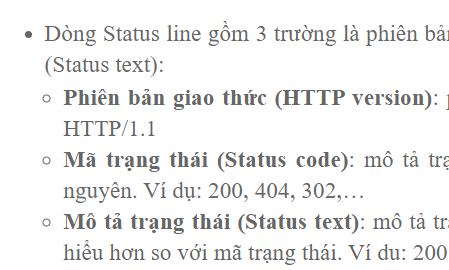
# Tìm hiểu HTTP, HTTPS?

HTTP là tên viết tắt của HyperText Transfer Protocol (*giao thức truyền tải siêu văn bản*), là một giao thức cơ bản dùng cho World Wide Web (*www*) để truyền tải dữ liệu dưới dạng văn bản, hình ảnh, video, âm thanh và các tập tin khác từ Web server đến các trình duyệt web và ngược lại.









HTTPS là viết tắt của Hypertext Transfer Protocol Secure. Vấn đề với giao thức HTTP thông thường là thông tin từ máy chủ đến trình duyệt không được mã hóa, có nghĩa là nó có thể dễ dàng bị đánh cắp . Giao thức HTTPS khắc phục điều này bằng cách sử dụng chứng chỉ SSL (lớp cổng bảo mật) , giúp *tạo kết nối được mã hóa an toàn* giữa máy chủ và trình duyệt, từ đó bảo vệ thông tin nhạy cảm bị đánh cắp khi thông tin được chuyển giữa máy chủ và trình duyệt.

HTTP hoạt động trên mô hình Client (máy khách) –Server (máy chủ). Các máy khách sẽ gửi yêu cầu đến máy chủ và chờ sự hồi đáp của máy chủ. Để có thể trao đổi thông tin được với nhau, các máy chủ và máy khách phải thực hiện trên một giao thức thống nhất, đó chính là HTTP.

HTTPS hoạt động tương tự như HTTP nhưng được bổ sung thêm chứng chỉ SSL (Secure Sockets Layer. SSL là tiêu chuẩn của công nghệ bảo mật, truyền thông mã hoá giữa máy chủ Web server và trình duyệt) và giao thức TLS (**Transport Layer Security sử dụng private key,** TLS là một phiên bản mới của giao thức SSL). Các giao thức này đảm bảo rằng không ai khác ngoài các máy khách và máy chủ có thể hack thông tin, dữ liệu ra ngoài.

Chạy local là http, khi deploy lên thì sẽ là https

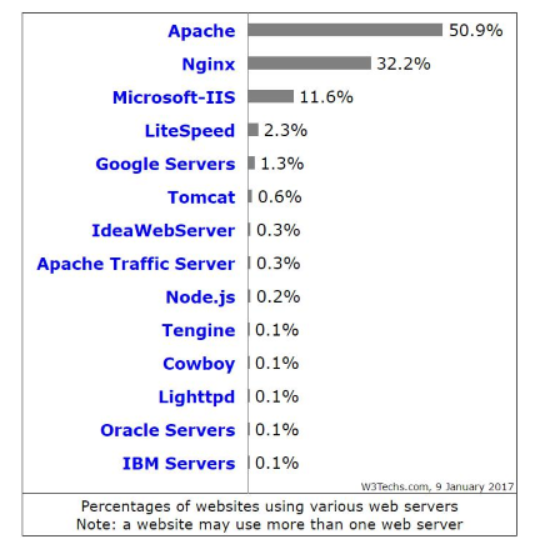
* HTTP sử dụng số cổng 80 để giao tiếp và sử dụng HTTPS 443
* HTTP được coi là không an toàn và HTTPS an toàn
* HTTP hoạt động ở tầng ứng dụng và HTTPS hoạt động ở tầng giao vận
* HTTP không yêu cầu bất kỳ chứng chỉ nào và HTTPS cần chứng chỉ SSL

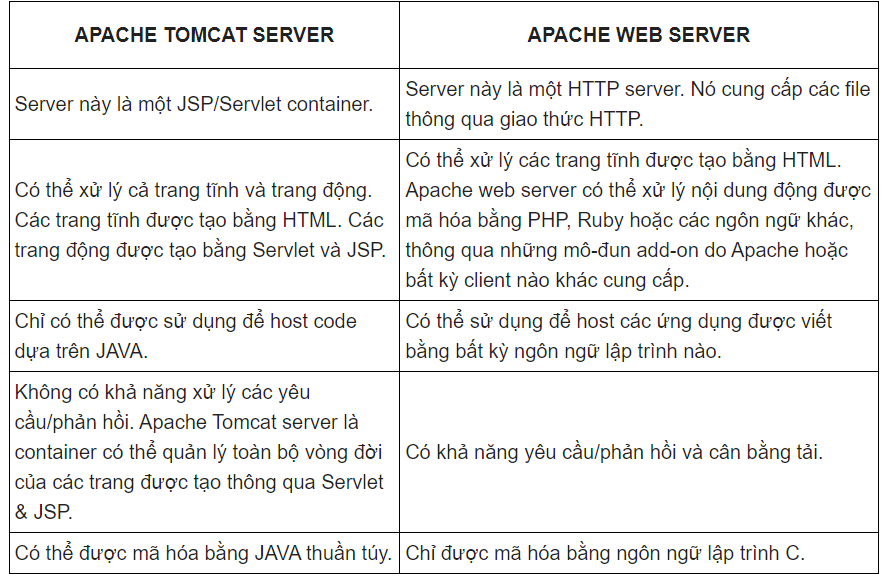
### **Các chuyển đổi sang HTTPS**

1. Mua chứng chỉ SSL và địa chỉ IP chuyên dụng từ các công ty cung cấp host cho bạn
2. Cài đặt và định cấu hình chứng chỉ SSL.
3. Thực hiện sao lưu toàn bộ trang web của bạn trong trường hợp bạn cần trở lại trạng thái ban đầu
4. Cấu hình bất kỳ liên kết hard internal nào trong trang web của bạn, từ HTTP sang HTTPS.
5. Cập nhật các thư viện code của bạn, chẳng hạn như JavaScript, Ajax và bất kỳ plugin của bên thứ ba nào.
6. Chuyển hướng mọi liên kết bên ngoài mà bạn kiểm soát sang HTTPS, chẳng hạn như danh sách thư mục.
7. Cập nhật các [htaccess](https://httpd.apache.org/docs/current/howto/htaccess.html), như Apache Web Server, [LiteSpeed](https://www.litespeedtech.com/), [NGinx Config](http://nginx.org/en/docs/beginners_guide.html) và chức năng quản lý dịch vụ internet của bạn (chẳng hạn như Windows Web Server), để redirect từ HTTP sang HTTPS.
8. Nếu bạn đang sử dụng [CDN](https://en.wikipedia.org/wiki/Content_delivery_network), hãy cập nhật cài đặt SSL của CDN.
9. Triển khai [301 redirects](https://moz.com/learn/seo/redirection) trên từng trang.
10. Cập nhật mọi liên kết bạn sử dụng trong các tool tự động hóa marketing, chẳng hạn như liên kết email.
11. Cập nhật bất kỳ trang đích và liên kết tìm kiếm nào có trả tiền.
12. Thiết lập trang web HTTPS trong [Google Search Console](https://search.google.com/search-console/about?hl=en&utm_source=wmx&utm_medium=wmx-welcome) và [Google Analytics](https://marketingplatform.google.com/about/analytics/).

* Web server là gì?

**Web server** là máy chủ cài đặt các chương trình phục vụ các ứng dụng web. Webserver có khả năng tiếp nhận request từ các trình duyệt web và gửi phản hồi đến client thông qua giao thức HTTP hoặc các giao thức khác.





### **Trình duyệt phân giải tên miền thành địa chỉ IP**

Trình duyệt web của bạn trước tiên cần phải xác định địa chỉ IP nào mà tên miền topdev.vn trỏ về. Trình duyệt sẽ yêu cầu thông tin từ một hoặc nhiều máy chủ DNS (thông qua internet, DNS là viết tắt của cụm từ Domain Name System, mang ý nghĩa đầy đủ là hệ thống phân giải tên miền.). Máy chủ DNS sẽ cho trình duyệt biết địa chỉ IP nào tên miền sẽ trỏ đến cũng là nơi đặt trang web.

Lúc này trình duyệt web đã biết địa chỉ IP của trang web, nó có thể yêu cầu URL đầy đủ từ webserver.

# Cổng (PORT) trong backend và frontend là gì?

Port là giao thức 16 bit đứng đầu (chèn vào phần đầu header) của mỗi gói tin trong giao thức TCP, [UDP](https://bkhost.vn/blog/giao-thuc-udp/) hay còn gọi là cổng port, nơi quy định các tập [dữ liệu](https://bkhost.vn/blog/data-du-lieu/) riêng biệt. Port là một dạng thuật toán đã được định sẵn và mỗi máy tính cần phải có thì mới có thể nhận và gửi các gói tin đi được. Khi hoạt động Port cũng được quy đổi giống với số bit của bất kì một mã dữ liệu nào đó. Đơn giản thì đây giống như cánh cổng có quyền cho vào hay không với các dữ liệu muốn ra vào hệ thống máy tính của bạn.

Cần 1 [địa chỉ IP](https://bkhost.vn/blog/dia-chi-ip/), bạn xác định được một máy trong một mạng, nhưng nếu chúng chạy cùng nhiều dịch vụ khác nhau thì phải có cách để phân biệt được chúng và lúc này bạn cần đến Port.

* Phân tích Database?

Lớp thuộc môn học tương ứng, câu hỏi trong bài kiểm tra trắc nghiệm sao lại để riêng để mở rộng chức năng tự động đề ngân hàng câu hỏi sau này, bảng vai trò với user quan hệ nhiều nhiều nên cần có bảng ngoại lai.

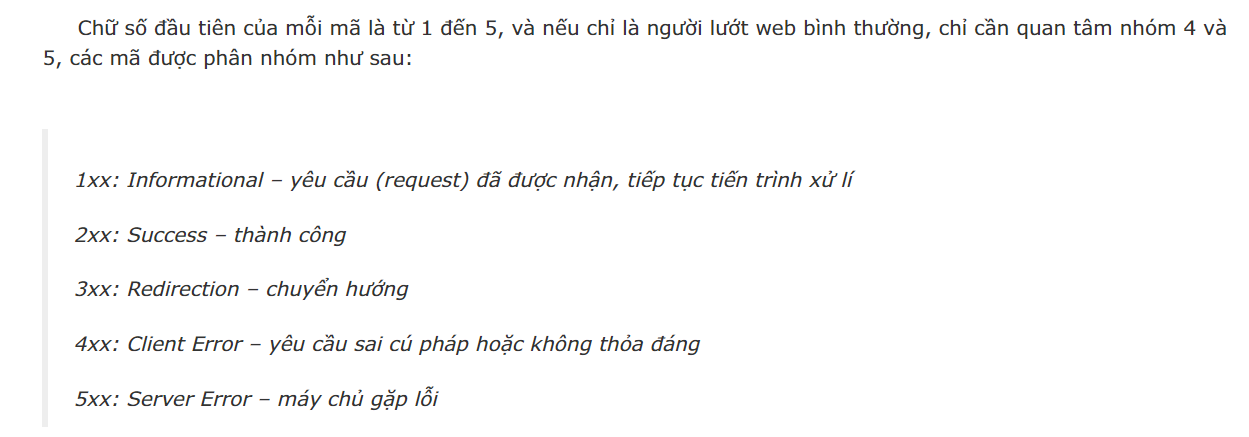
* Nguyên lý hoạt động của hệ thống?

FE gửi yêu cầu thông qua APIs, BE lọc lấy APIs đó có quyền truy cập hay không và thực hiện yêu cầu nếu thỏa đáng quyền truy cập.

# Phân biệt Authentication, authorization?

Authentication (xác thực) có nghĩa là xác nhận danh tính của riêng bạn, trong khi authorization (ủy quyền) có nghĩa là cấp quyền truy cập vào hệ thống. Nói một cách đơn giản, authentication là quá trình xác minh bạn là ai, trong khi authorization là quá trình xác minh những gì bạn có quyền truy cập.

* Các mã mà Backend có thể gửi về? (200. 404, 500)

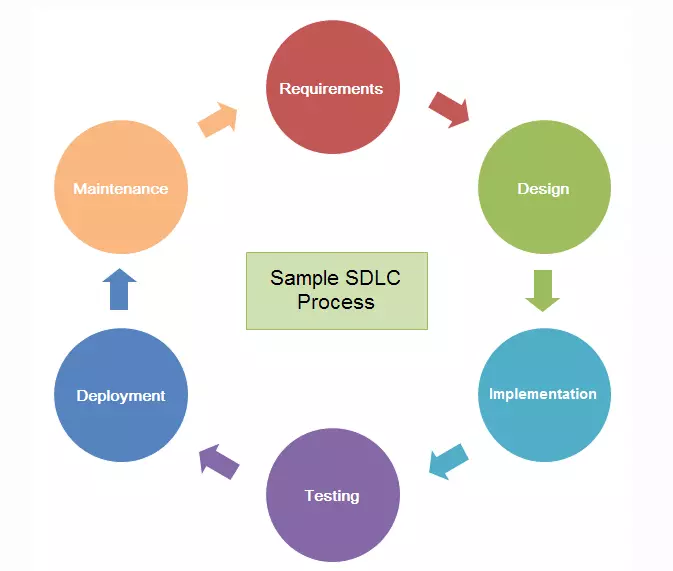


* Các ứng dụng, hướng phát triển của sản phẩm?

ứng dụng vào cho các trường học, đại học khi team phải mua bản quyền, hay gg cần phải tạo tài khoản với nhiều bước rắc rối.

* Sơ đồ lớp, sơ đồ tuần tự, sơ đồ, include, extend trong sơ đồ use case biểu thị điều gì?

# Quy trình vòng đời phần mềm?

Vòng đời phát triển phần mềm/ software development life-cycle (SDLC): Mô hình thác nước (waterfall)  


* *Requirment Analysis (Thu thập, phân tích yêu cầu):* đây phần khá quan trong trong việc phát triển phần mềm. Thông thường khi bắt đầu làm sản phẩm, khách hàng thường không định hình rõ được mình muốn làm một hệ thống như thế nào nên yêu cầu đưa ra sẽ không đầy đủ, không rõ ràng hoặc mâu thuẫn lẫn nhau. Lúc này, đội phát triển bao gồm những kỹ sư có kinh nghiệm sẽ xác nhận/ góp ý/ sửa đổi để có một yêu cầu cơ bản hoàn thiện ban đầu.
* *Design (Thiết kế):* Kiến trúc hệ thống liên quan đến việc bảo đảm rằng hệ thống phần mềm sẽ đáp ứng đầy đủ các yêu cầu của sản phẩm, cũng như đảm bảo rằng các yêu cầu trong tương lai có thể được giải quyết. Nó cũng liên quan đến việc giao tiếp giữa các hệ thống phần mềm và các sản phẩm phần mềm khác, cũng như các phần cứng cơ bản hoặc các hệ điều hành chủ.
* *Implementation (Thực hiện/xây dựng):* Thiết kế trước đó phải được lập trình viên dịch sang một dạng máy tính có thể đọc và hiểu được. Nếu thiết kế được thực hiện một cách chi tiết và đầy đủ thì đến giai đoạn này việc code sẽ rất dễ dàng. Ngược lại, nếu trước đó việc thiết kế hệ thống quá sơ sài thì đến giai đoạn này sẽ gặp nhiều khó khăn và mất thời gian giải quyết.
* *Testing (Kiểm thử)*: Sau khi các lập trình viên đã hoàn thành việc phiên dịch mã code, đội kiểm thử viên bắt đầu công việc của mình. Đội ngũ kiểm thử sẽ dùng các phương pháp kiểm thử khác nhau để phát hiện ra những lỗi trên hệ thống, trong giai đoạn này, các công cụ kiểm thử tự động, các tool hỗ trợ cũng sẽ được sử dụng nhằm phát hiện ra lỗi để đội phát triển có thể kịp thời sửa chữa trước khi đến với người dùng cuối. Ngoài ra, hiện nay cũng có nhiều công ty tự xây dựng công cụ kiểm thử để phục vụ cho các hoạt động phát triển của họ.
* *Deployment (Triển khai):* Sau khi đội kiểm thử kết thúc công việc của mình, sản phẩm đảm bảo có thể đưa vào sử dụng thì nó sẽ được đưa vào sử dụng trong thực tế.
* *Maintenance (Bảo trì):* Bảo trì và nâng cấp phần mềm để đối phó với các vấn đề được phát hiện hoặc yêu cầu mới có thể tốn nhiều thời gian hơn so với việc phát triển ban đầu của phần mềm.

# UDP, TCP là gì?

| STT | Đặc điểm | TCP (Transmission Control Protocol) | UDP (User Datagram Protocol) |
| --- | --- | --- | --- |
| 1 | Định nghĩa | Là một giao thức truyền thông tin, dùng dữ liệu được truyền giữa các hệ thống với nhau qua mạng. Tại đó, dữ liệu được truyền thông qua hình thức packet. Nó bao gồm xác minh lỗi, có nhiệm vụ phân phối và duy trì trình tự của các packet. | Tương tự như TCP. Nhưng UDP chịu trách nhiệm việc kiểm tra lỗi cùng khôi phục dữ liệu. Người dùng khi sử dụng giao thức này, dữ liệu sẽ được gửi không ngừng, không ghi nhận vấn đề ở đầu nhận. |
| 2 | Thiết kế | Ở TCP có định hướng kết nối (connection oriented). | UDP là một giao thức kém kết nối hơn (connectionless). |
| 3 | Độ tin cậy | Xuất phát từ việc TCP thực hiện hỗ trợ kiểm tra lỗi cùng cũng chịu trách nhiệm cung cấp dữ liệu đầu nhận. Điều này khiến cho TCP trở nên tin cậy hơn so với UDP. | Đối với, UDP chỉ thực hiện hỗ trợ kiểm tra lỗi cơ bản nhất thông qua checksum. Vì vậy UDP sẽ không đảm bảo việc phân phối dữ liệu tại điểm nhận. |
| 4 | Truyền dữ liệu | Tại giao thức TCP, dữ liệu được truyền theo một thứ tự xác định. Đồng nghĩa với các packet đến người nhận theo một trình tự. | Ở UDP, dữ liệu không được truyền theo thứ tự. Sẽ có một lớp ứng dụng đảm nhiệm sắp xếp. |
| 5 | Hiệu năng | TCP được đánh giá là chậm hơn và hiệu suất không bằng UDP. Đồng thời, TCP cũng nặng hơn so với UDP. | Ngược lại với tới TCP, UDP nhanh hơn và hiệu quả hơn rất nhiều. |
| 6 | Truyền lại | Packet có thể được gửi lại nếu TCP gói bị mất hoặc cần phải gửi lại. | Không được truyền lại các packet trong UDP. |

* Lỗ hổng bảo mật SQL injection?

không kiểm tra đầu vào

lỗi bảo mật trên máy chủ

=> thay đổi dữ liệu làm sai lệch cơ sở dữ liệu

* Phân biệt Local storage, session storage, Cookies?

LocalStorage giúp website lưu trữ dữ liệu vĩnh viễn trên trình duyệt. Trừ khi người dùng xóa cache hoặc làm các hành động như cài lại trình duyệt.

sessionStorage khá giống với localStorage. Vì chúng đều thuộc về web storage API.

* Lưu data theo cặp key/value ở local browser và phía server không access được các data này.
* Có cùng APIs: setItem, getItem, removeItem, clear.
* Cho phép lưu trữ nhiều data(khoảng 10MB).

Một khuyết điểm của cả localStorage và sessionStorage là có thể bị đọc bởi Javascript. Do đó dễ bị đánh cắp thông tin thông qua một cross-site scriting.

* localStorage chỉ access được trên browser client; còn cookies thì có thể access được ở browser client và cả phía server (khi tạo một http request thì cookies của browser sẽ được attach vào header Cookie, từ đó phía server có thể parse header này và get được data cookie).
* cookies có thời gian hết hạn expires, sau thời gian này thì cookies sẽ biến mất khỏi browser.
* cookies chỉ cho phép lưu tối đa khoảng 4 KB, vì vậy ta nên sử dụng cookies với mục đích save những loại data simple ví dụ như token cho authentication,...
* Tại sao lại lựa chọn Spring boot?
* Reactjs thiết kế theo mô hình SPA? Phân biệt SPA và MPA?

Single-page Application là một ứng dụng hoạt động trên browser và không cần bắt buộc phải reload lại khi sử dụng. Lúc đó client render những đoạn mã HTML, CSS được gọi là Client Side Rendering. Thiết kế FE, BE có thể diễn ra song song.

Với MPA, chúng ta có một phong cách hiển thị rất "truyền thống" là mọi dữ liệu sẽ được render ra từ mã HTML, CSS và cả Javascript và trả lại về người dùng hay được gọi là Server Side Rendering. Thiết kế FE, BE diễn ra cũng lúc.

* Sản phẩm có gì ưu điểm hơn so với những sản phẩm đã có trên thị trường? (như của gg, team, …)

Giá thành hợp lý

Không cần đăng ký phức tạp

Có thể dễ dàng kiểm soát tài khoản

Xem được kết quả các bài kiểm tra trắc nghiệm

* Tìm hiểu về spring security?
* Hàm băm SHA để mã hóa mật khẩu?

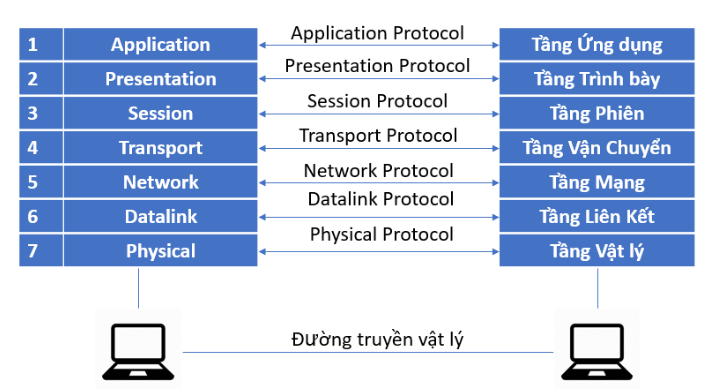
Ví dụ là SHA-2: chuyển về nhị phân, thêm số 1 ở cuối thêm số 0 cho đến khi đủ 512 bit, Nối 64 bit vào cuối (là độ dài của dữ liệu ban đầu)

SHA 512

* URL là gì?

**URL** là chữ viết tắt của “**Uniform Resource Locator**“, tạm dịch: “*Trình định vị tài nguyên thống nhất*“. Đây là một đường dẫn liên kết đến website.

* Mô hình OSI?



### **Giao thức hướng liên kết (Connection Oriented)**

### **Giao thức không liên kết (Connectionless)**

HTTP ở tầng ứng dụng

Tầng ứng dụng là lớp trên cùng, xác định giao diện giữa người sử dụng và môi trường OSI. Tầng ứng dụng được sử dụng bởi phần mềm người dùng cuối như trình duyệt web và ứng dụng email. Nó cung cấp các giao thức cho phép phần mềm gửi, nhận thông tin và trình bày dữ liệu có ý nghĩa cho người dùng.

Tầng thứ hai kế tiếp tầng ứng dụng là tầng trình bày, tầng này sẽ giải quyết các vấn đề liên quan đến các cú pháp và ngữ nghĩa của thông tin được truyền.

Tầng phiên là lớp chịu trách nhiệm đóng mở giao tiếp giữa hai thiết bị.

Tầng vận chuyển chịu trách nhiệm giao tiếp đầu cuối giữa hai thiết bị. Điều này bao gồm việc lấy dữ liệu từ lớp phiên và chia nó thành các phần được gọi là phân đoạn trước khi gửi đến tầng mạng.

Tầng mạng có nhiệm vụ tạo điều kiện thuận lợi cho việc truyền dữ liệu giữa hai mạng khác nhau.

Tầng liên kết dữ liệu rất giống với tầng mạng, ngoại trừ tầng liên kết dữ liệu tạo điều kiện thuận lợi cho việc truyền dữ liệu giữa hai thiết bị trên cùng mạng.

Lớp này bao gồm các thiết bị vật lý liên quan đến việc truyền dữ liệu, chẳng hạn như cáp và thiết bị chuyển mạch.

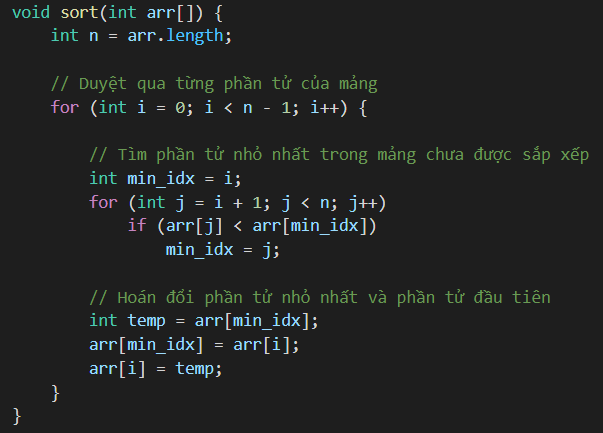
* Chỗ này dùng cho mục đích gì, hoạt động như nào, cải tiến được không, đây là theo quan điểm cá nhân hay có nguyên cứu chứng minh gì?

# Sắp xếp

## +) Sắp xếp lựa chọn (selection sort)

Mảng sẽ được chia thành 2 phần: phần được sắp xếp cố định, phần chưa được sắp xếp. Ban đầu, phần được sắp xếp là trống và phần chưa được sắp xếp là toàn bộ danh sách ban đầu. Nếu sắp xếp theo thứ tự tăng dần thì phần tử nhỏ nhất trong mảng sẽ được đổi chỗ cho phần tử đầu tiên và trở thành phần tử của phần đã sắp xếp. Làm tương tự đến hết.

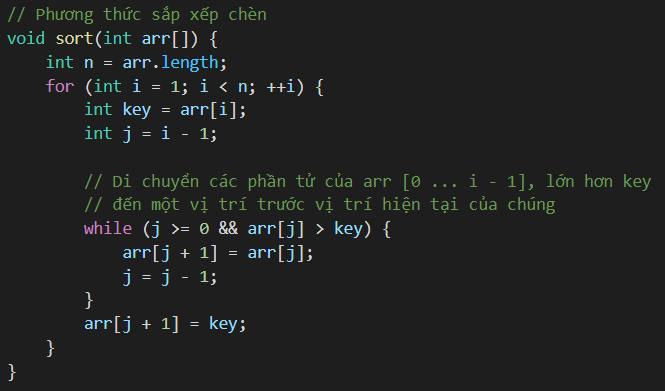
Độ phức tạp: O(n^2)



## +) Sắp xếp chèn (insertion sort)

Mảng cũng được chia thành 2 phần đã sắp xếp không cố định và chưa sắp xếp. Xét từ vị trí phần tử thứ 2 của mảng và chèn nó vào vị trí theo chiều được yêu cầu. Tiếp tục thực thi tương tự với các phần tử còn lại.

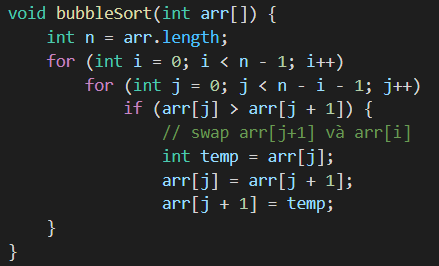
Độ phức tạp: O(n^2)



## +) Sắp xếp nổi bọt (bubble sort)

Mảng được chia thành 2 phần 1 phần cố định đã được sắp xếp và phần chưa được sắp xếp. Thuật toán sẽ thực hiện từ đầu đến cuối các phần tử liên tiếp sẽ so sánh với nhau và nếu phân tử ở trước lớn hơn phần tử ở sau sẽ được đổi 2 phần tử liên tiếp đó và phần tử cuối cùng sẽ là phần tử lớn nhất. Tiếp tục thực hiện với phần mảng đã bỏ phần tử cuối cùng.

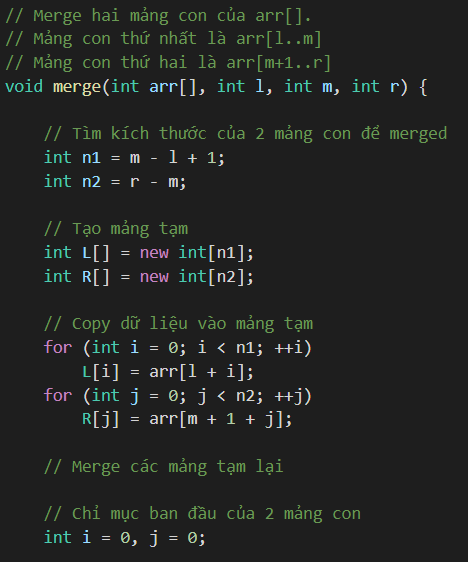
Độ phức tạp: O(n^2)

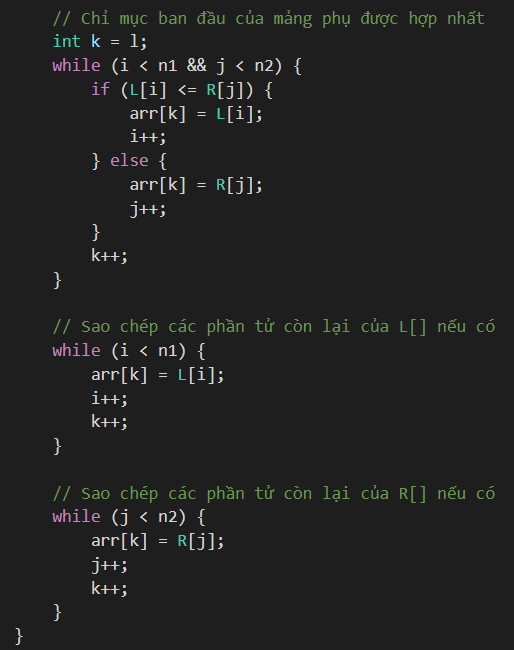


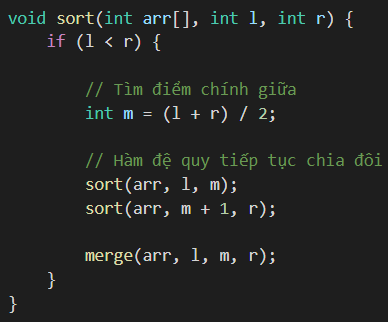
## +) Sắp xếp trộn (hòa nhập) (merge sort)

Chia đôi mảng ban đầu rồi tiếp tục chia đôi cho đến khi chỉ còn 2 phần tử để dễ so sánh. Sau đó ta tiến hành gộp những phần đã được sắp xếp vào với nhau theo thứ tự.

Độ phức tạp: O(nlog(n))



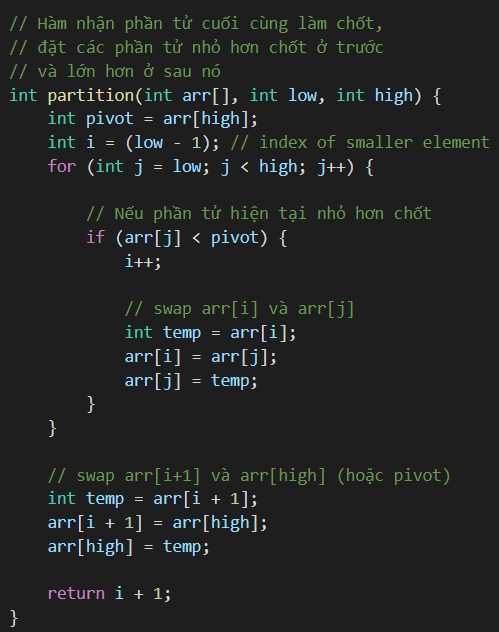


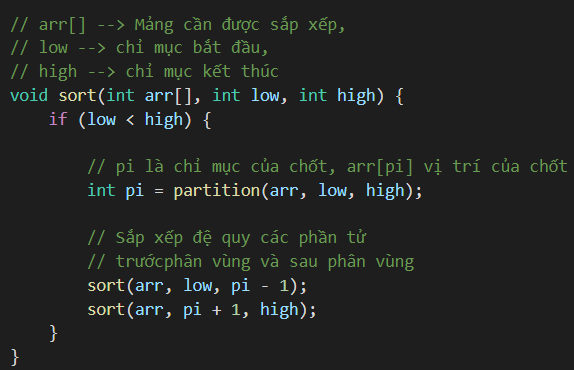


## +) Sắp xếp nhanh (quick sort)

Chọn 1 phần tử làm chốt tránh gây ra lặp vô hạn (có thể chọn phần tử đầu, cuối, ở giữa hoặc phần tử bất kỳ). Chọn phần tử ở cuối làm chốt, xét tất cả phần tử bé hơn hoặc bằng chốt thì chuyển lên đầu rồi cuối cùng đổi chỗ chốt với phần tử tiếp theo của phần tử mới nhất được đổi chỗ. Làm tương tự cho đến khi còn 1 phần tử.

Độ phức tạp: O(nlog(n))

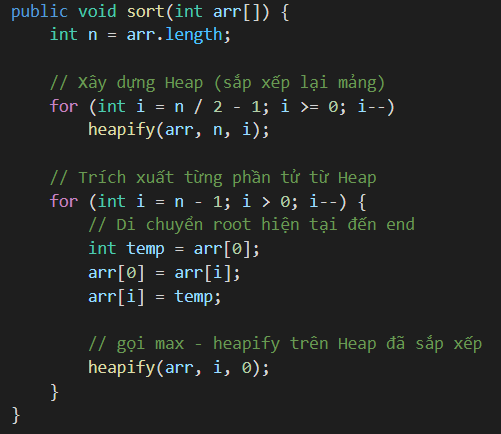




## +) Sắp xếp vun đống (heap sort)

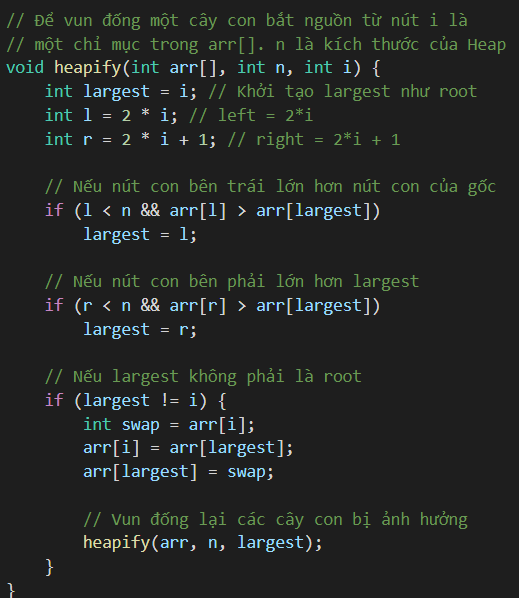
Bước đầu tạo thành cây với nút con tại đỉnh sẽ lớn hơn các nút con của nó bằng cách xét từ nút thứ n/2 - 1. Đổi vị trí root cho phần tử cuối cùng và thực hiện heapify với nút root mới để tìm ra phần tử lớn tiếp theo để chuyển về vị trí trước vị trí lúc nãy đã chuyển về.

Độ phức tạp: O(nLogn).



Chỗ l = 2 \* i + 1;

r = l + 1;



# Tìm kiếm

## +) Thuật toán tìm kiếm tuần tự (Linear Search)

Xét từ đầu đến cuối

Độ phức tạp: O(n)

## +) Thuật toán cây nhị phân tìm kiếm (Binary search tree)

Nút con bên trái nhỏ hơn nút con tại đỉnh

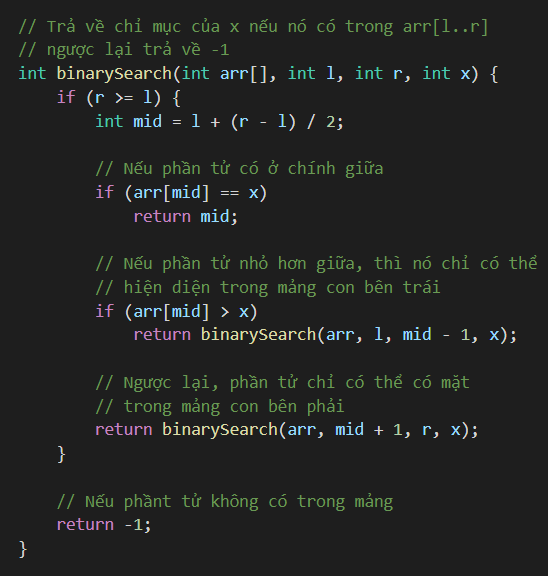
Nút con bên phải lớn hơn nút con tại đỉnh

mỗi cây con của 1 nút là một cây nhị phân tìm kiếm

## +) Thuật toán tìm kiếm nhị phân (Binary Search)

Yêu cầu mảng đầu vào đã được sắp xếp

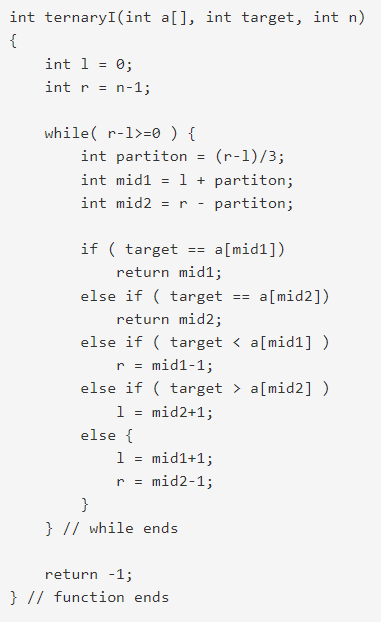
Độ phức tạp: O (log [n]) trong đó cơ số của log = 2



## +) Thuật toán tìm kiếm bậc ba (Ternary Search)

Cũng giống nhị phân vẫn cần mảng đó đã được sắp xếp

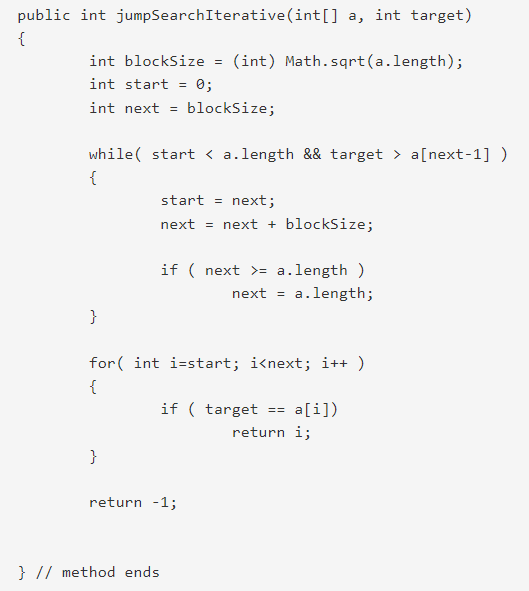
Độ phức tạp: O (log [n]) trong đó cơ số của log = 3



## +) Thuật toán tìm kiếm nhảy (Jump Search)

Áp dụng với mảng đã được sắp xếp. Thực hiện tìm kiếm trong một khoảng có độ dài bằng căn bậc 2 của số lượng phần tử mảng.

Độ phức tạp: O(log(sqrt(n))

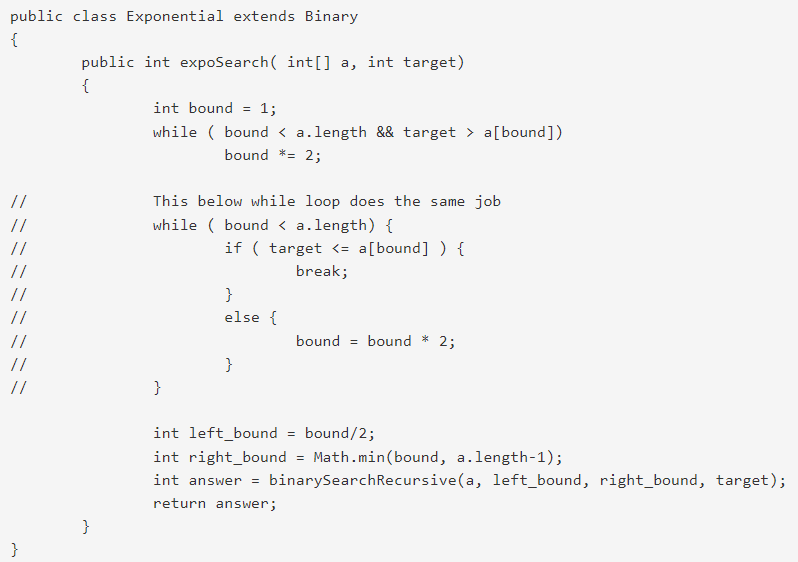


## +) Thuật toán tìm kiếm cấp số nhân (Exponential Search)

Áp dụng với mảng đã được sắp xếp.

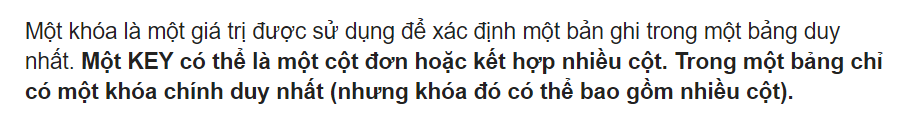
Thay vì tìm kiếm nhị phân trên toàn bộ mảng thì ta sẽ thực hiện tìm kiếm trong một giới hạn xác định nhỏ hơn. Ta thực hiện tăng gấp đôi chỉ số giới hạn của mảng từ 1 cho đến khi số cần tìm bé hơn giá trị của phần tử tại chỉ số đó, rồi ta áp dụng thuật toán tìm kiếm nhị phân với giới hạn trái bằng chỉ số đó chia 2, giới hạn phải bằng chỉ số đó hoặc phần tử cuối cùng của mảng.

Độ phức tạp: O (log [i]) số phần tử nằm trong mảng cần tìm.



# Các chuẩn của chuẩn hóa dữ liệu

+) Khóa chính:

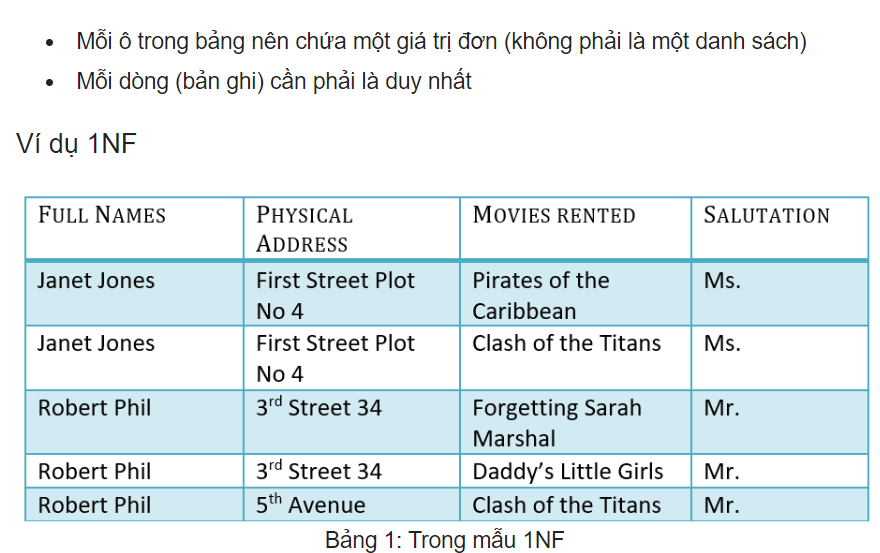


+) Khóa phụ:

Khóa ngoại của một table được xem như con trỏ trỏ tới khóa chính của table khác

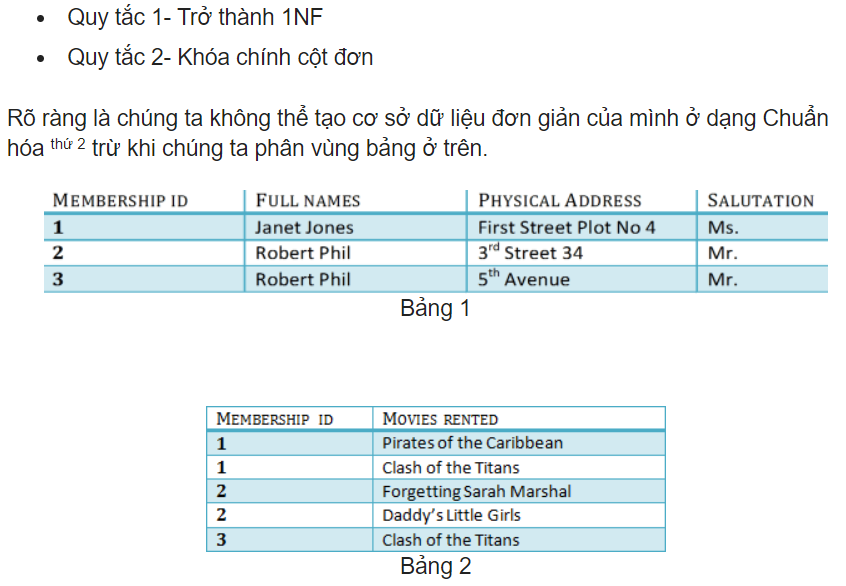
+) Bảng mapping: là những liên kết của các bảng trong cơ sở dữ liệu.

## +) Quy tắc 1NF:



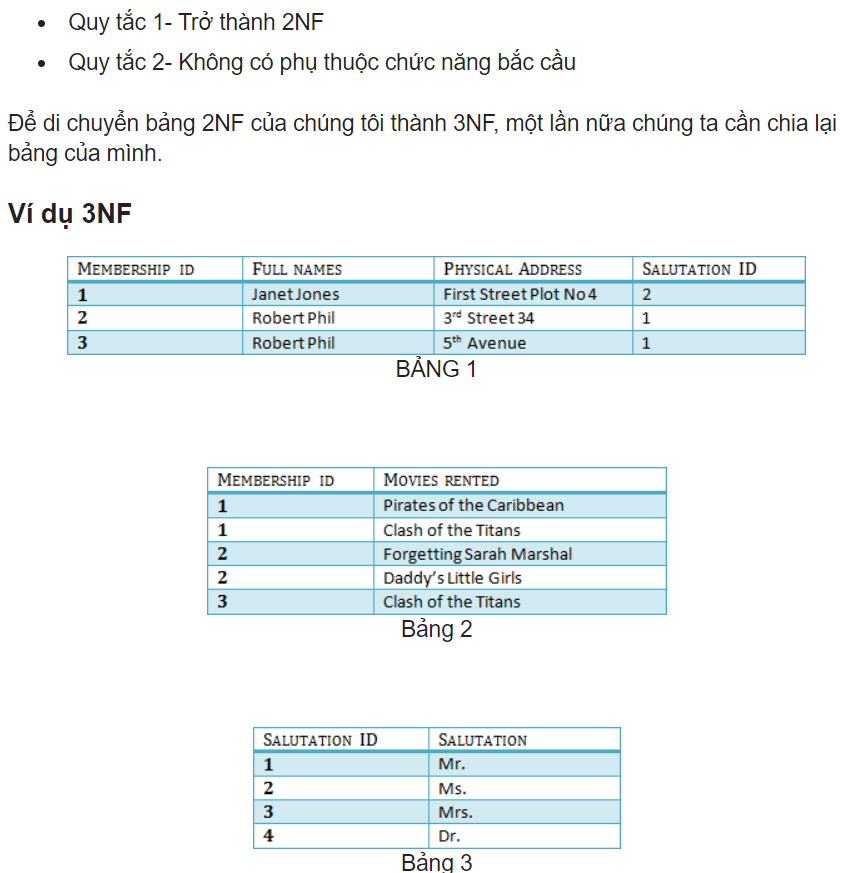
## +) Quy tắc 2NF:

Các cột không khóa phải phụ thuộc duy nhất vào cột khóa chính.



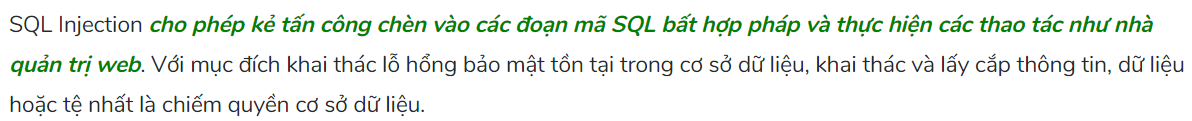
## +) Quy tắc 3NF:

Chức năng bắc cầu là cột không khóa này thay đổi có thể khiến cột không khóa khác thay đổi theo. Ví dụ như quê quán chuyển từ Hà Nội thành hà Nội 1 thì phải đi sửa hết nên tách riêng ra 1 bảng => Để sử dụng khóa ngoại lai.



## +) Quy tắc BCNF:

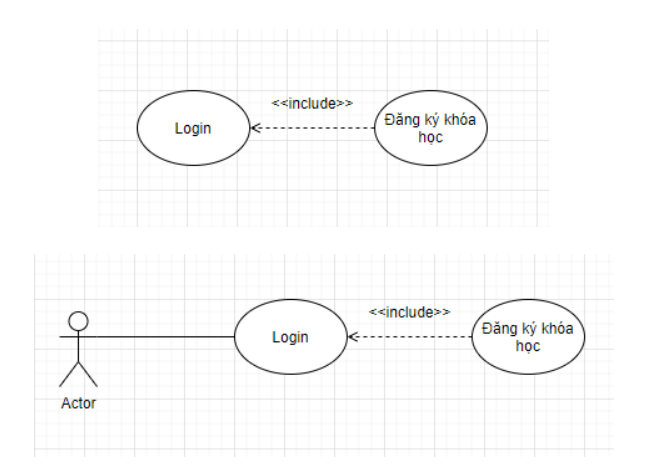
# SQL injection



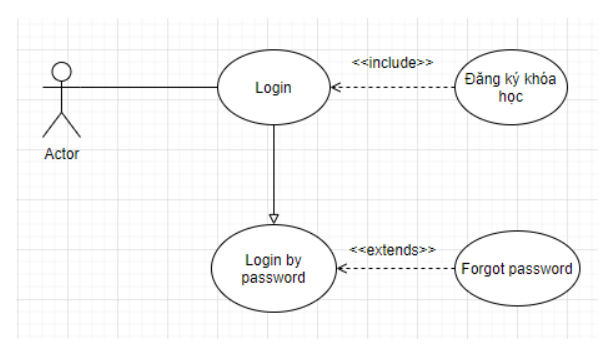
* Nguyên nhân: không kiểm tra, lọc dữ liệu đầu vào trước khi thực thi câu lệnh SQL.
* Ví dụ:
  + [Truy xuất dữ liệu ẩn](https://portswigger.net/web-security/sql-injection#retrieving-hidden-data) , nơi bạn có thể sửa đổi truy vấn SQL để trả về kết quả bổ sung. Ví dụ: như muốn xem sản phẩm chưa phát hành thay vì truyền release = 1 thì kẻ tấn công sẽ truyền release = 0.
  + [Subverting logic của ứng dụng](https://portswigger.net/web-security/sql-injection#subverting-application-logic) , nơi bạn có thể thay đổi một truy vấn để can thiệp vào logic của ứng dụng. Ví dụ: khi login thì SQL sẽ tìm những tài khoản có tên đăng nhập và mật khẩu trong bảng tài khoản, kẻ tấn công có thể chỉ truyền tên đăng nhập và bỏ qua mật khẩu thì vẫn có thể đăng nhập thành công khi máy chủ không chủ động lọc dữ liệu.
* Cách phòng tránh
  + Sử dụng câu lệnh SQL đã được tham số hóa.
  + Lọc bỏ những ký tự đặc biệt tại đầu vào của câu SQL.
  + Phân quyền cho các loại tài khoản riêng biệt.

# So sánh include và extend trong use case

* Include: Trong use case, Include được định nghĩa là mối quan hệ bắt buộc phải có giữa các Use Case với nhau. Use Case A có mối quan hệ với Use Case B, điều đó có nghĩa Use Case A bao gồm Use Case B. Để Use Case A xảy ra thì phải đạt được Use Case B. Mũi tên nét đứt hướng về use case bắt buộc. Ví dụ:



* Extend: Extend biểu diễn mối quan hệ mở rộng giữa các Use Case với nhau. Nếu Include thể hiện mối quan hệ bắt buộc thì Extend lại là mối quan hệ không bắt buộc (có thể có hoặc không) giữa các Use Case với nhau. Nếu Use Case B là Extend của Use Case A, điều này có nghĩa Use Case B chỉ là một optional chỉ xảy ra trong một hoàn cảnh cụ thể nào đó. Mũi tên nét đứt hướng về phía use case cha.



# Sơ đồ use case

Là các chức năng mà các actor sẽ sử dụng để thể hiện sự tương tác giữa những người dùng và hệ thống.

# Sơ đồ class

Mô tả thành phần của các đối tượng trong hệ thống và các loại quan hệ tồn tại giữa chúng.

# Sơ đồ sequence

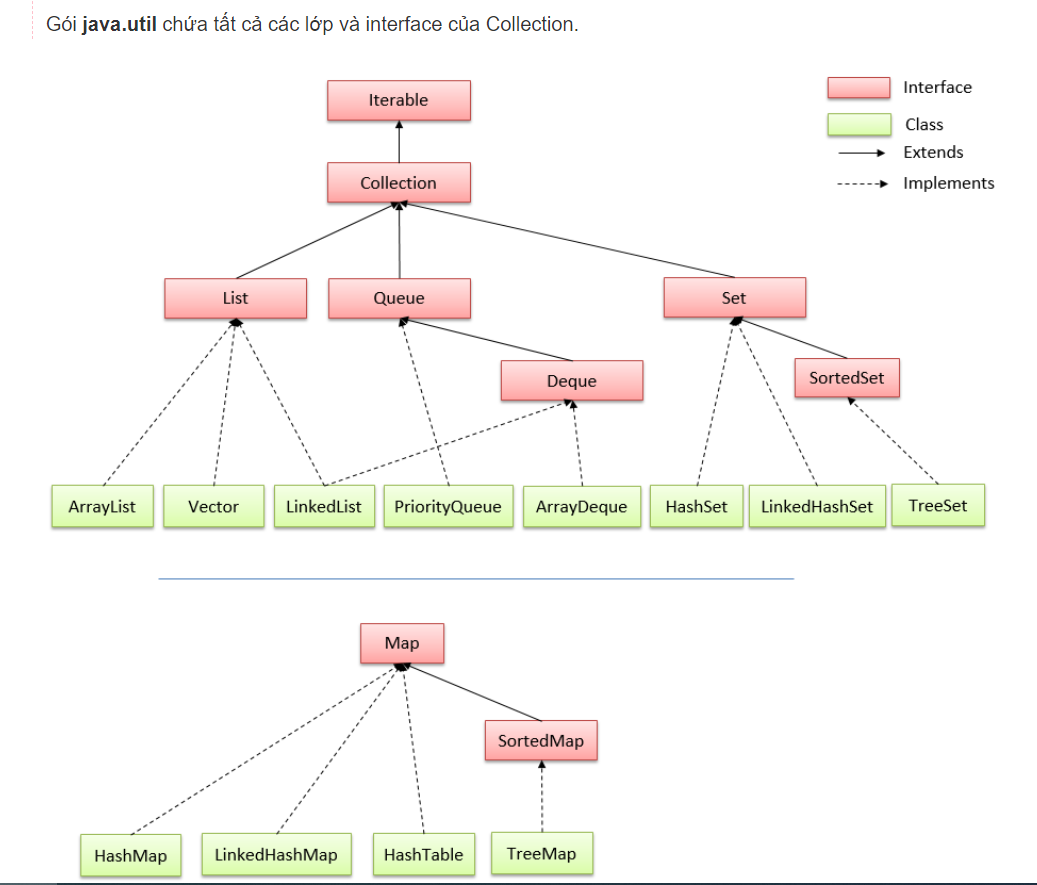
là biểu đồ dùng để xác định các trình tự diễn ra 1 sự kiện của một đối tượng nào đó.

# Sơ đồ activity

mô tả các hoạt động, luồng xử lý bên trong hệ thống.

# Câu hỏi

* Các anh có hỏi về cách tính độ phức tạp của mấy thuật toán khó như quick, merge, heap sort không? Có thể nhưng thường sẽ ít khi hỏi.



Công thức: n = a.n^2 + b.n + c

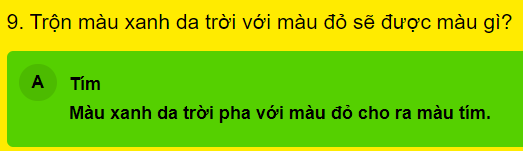
Nếu hiệu cấp 2 không đổi thì 2.a = hiệu cấp 2, 3.a + b = hiệu cấp 1 thứ nhất, a + b + c = số hạng thứ nhất.

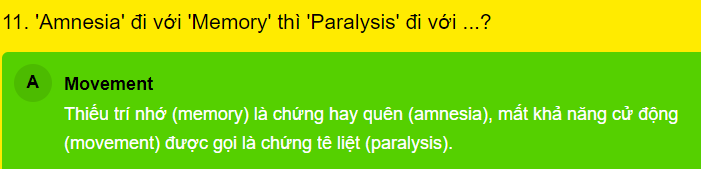
1. Sông nào chia cắt đàng trong đàng ngoài? Sông Gianh, sử sách hay gọi là Linh Giang, trở thành ranh giới chia nước Đại Việt thành Đàng Trong và Đàng Ngoài.
2. Tính đến năm 2020 nước nào chưa tham gia vào asean? Tính đến 2020, Đông Timor là nước duy nhất trong khu vực Đông Nam Á chưa gia nhập ASEAN.
3. Có 4 người A b c d qua sông người a đi mất 1 phút người b đi mất 2 p người c đi mất 7 phút, người d đi mất 10p hỏi thời gian ít nhất để đi qua cầu, mỗi lần đi chỉ đc đi 2 người? A,B đi cùng nhau mất 2p, A quay lại 1p, C,D đi qua cầu mất 10p B quay lại mất 2p, A,B qua cầu mất 2p -> mất 17p.
4. 3, 15, 35, 63, 99 điền số tiếp theo? 143
5. 5 3 3 35 3 4 4?
6. Bán rẻ mũ từ 30k xuống 20k anh trả 50k đổi tiền cho ông hàng xóm, hôm sau phát hiện tiền giả, hỏi lỗ bao tiền? 60k
7. Đất nước có diện tích lớn nhất thế giới? Nga
8. Đất nước có diện tích lớn thứ 2 thế giới? Canada
9. Đất nước đông dân nhất thế giới? Trung quốc
10. Nga có đường biên giới tiếp giáp với tất cả bao nhiêu nước? 14 nước.
11. Rộng hơn 2,7 triệu km vuông, Kazakhstan có diện tích lớn thứ 9 thế giới, đồng thời là quốc gia không giáp biển lớn nhất thế giới.
12. Brazil tên gọi chính thức là Cộng hòa Liên bang Brazil, chiếm khoảng một nửa diện tích của Nam Mỹ và là quốc gia lớn nhất Nam Mỹ.
13. Peru là nước lớn thứ ba Nam Mỹ.
14. Algeria là quốc gia rộng lớn nhất Châu Phi.
15. Greenland là hòn đảo lớn nhất thế giới.
16. Australia có diện tích lớn hơn Ấn Độ.
17. Quốc kỳ Algeria

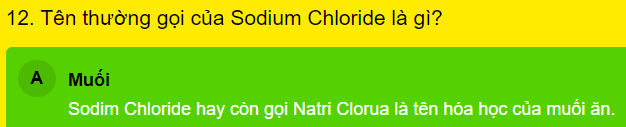


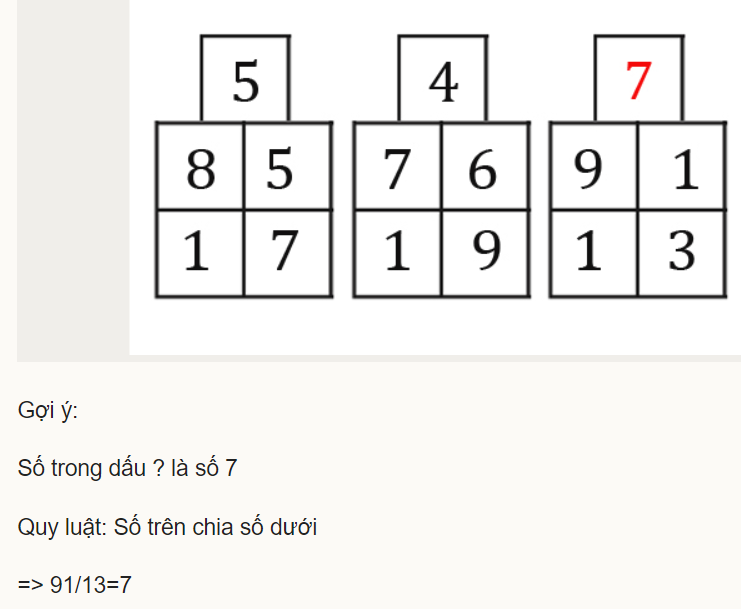
1. Ukraine sau Nga là quốc gia lớn nhất châu Âu.
2. Alaska là bang lớn nhất Hoa Kỳ.
3. Bangladesh không có chung đường biên giới với Trung Quốc.
4. Ai là người đầu tiên phát hiện Penicillin? Alexander Fleming
5. Mấy bài điền số vào tập hợp 3x4?
6. Từ khác loại với các từ còn lại? Gun, pistol,...











1. Các kiến trúc phần mềm, cách thức triển khai phần mềm? Waterfall(thác nước) box

* Mô hình thác nước – Waterfall model
* Mô hình chữ V – V- Shaped Model
* Mô hình xoắn ốc – Spiral Model
* Mô hình Agile – Agile Model
* Mô hình Scrum

1. Công đoạn mất thời gian nhất trong quá trình triển khai thác nước?
2. Union trong sql?
3. Tính độ phức tạp thuật toán? (ví dụ code dãy fibonacci)