1. Thuật toán 1

**public** List<Product> findAll() {

//{A}

List<Product> list = productDAO.findAll();

//{C}

E1

**for** (**int** i = 0; i < list.size() - 1; i++) {

//{C,E1}

E2

D

**for** (**int** j = i + 1; j < list.size(); j++) {

E3

**if** (list.get(i).getProductPrice() > list.get(j).getProductPrice()) {

Q1

P1

Q2

Collections.*swap*(list, i, j);

}

}//{C}

}//{C,!E1}

**return** list;

//{B}

}

* gọi mệnh đề thể hiện tính chất dữ liệu vào của chương trình {A} và mệnh đề  
  thể hiện tính chất dữ liệu ra cần có {B}, ta có:

+ {A: List<Product>, list.size() >0 } và {B: List<Product>, list.size() >0, list.get(0).getProductPrice() > list.get(1).getProductPrice(1) ... }

* Xét mệnh đề {C: List<Product>, list.size()>0 }
* Ta có: {A} List<Product> list = productDAO.findAll();{C}
* Để C là bất biến của Q1 trong điều kiện E1
* Ta cần có : {C,E1}Q2{C,E1} ({C,E1} là bất biến của Q2)
* Trong đoạn trình Q2:

+ Dự trù {C,E1} là bất biến của Q2 trong điều kiện E2

+ ta cần chứng minh {C,E1,E2,E3}P1{C}

+ Ta có: i, {C,E1,E2,E3}P1{C}

Ii, {C,E1,E2,!E3}=>L {C}

+ Vậy {C,E1} là bất biến của Q2, kết thúc Q2 ta có: {C}

* Kết thúc Q1 ta có: {C,!E1}
* Dễ dàng có được : {C1,!E1} =>L {B}

1. Thuật toán 2

**public** **boolean** save(Product product) {

{A}

E1

**if** (product.getProductId() == **null** || product.getProductId().isEmpty()) {

**throw** **new** Exception("Điền mã sản phẩm"); //**P1**

E2

} **else** **if** (product.getProductId().length() > 6) {

**throw** **new** Exception("Mã sản phẩm tối đa 6 ký tự");//**P2**

E3

} **else** **if** (product.getProductName() == **null**) {

**throw** **new** Exception("Điền tên sản phẩm");//**P3**

E4

} **else** **if** (!java.util.regex.Pattern.*matches*(REGEX\_NAME\_PRODUCT, product.getProductName())) {

**throw** **new** Exception("Tên sản phẩm từ 1 - 50 ký tự và chỉ chứa ký tự chữ");//**P4**

E5

} **else** **if** (product.getProductPrice() <= 0) {

**throw** **new** Exception("Giá sản phẩm phải lớn hơn 0");//**P5**

E6

} **else** **if** (!java.util.regex.Pattern.*matches*(REGEX\_UNIT\_PRODUCT, product.getProductUnit())) {

**throw** **new** Exception("Đơn vị tính sản phẩm chỉ chứa ký tự chữ");//**P6**

E7

} **else** **if** (!typeImage.contains(product.getProductImage()

.substring(product.getProductImage().lastIndexOf(".") + 1)

)) {

**throw** **new** Exception("Định dạng file chấp nhận " + typeImage); //**P7**

}

E8

**if** (productDAO.findById(product.getProductId()) != **null**) {

**throw** **new** Exception("Sản phẩm đã tồn tại");//**P8**

}

**return** productDAO.save(product);

{B}

}

* Gọi mệnh đề thể hiện tính chất dữ liệu vào của chương trình {A} và mệnh đề  
  thể hiện tính chất dữ liệu ra cần có {B}, ta có:

+ {A: Product, } và {B: Product, thông báo lỗi }

* Ta cần chứng minh {A}P{B}
* Áp dụng tính chất cấu trúc phép chọn ta có:
* C2:

+ 1,{A, E1}P1{B}

+ 2,{A, !E1, E2}P2{B}

+ 3,{A, !E1, !E2, E3}P3{B}

+ 4,{A, !E1, !E2, !E3, E4}P4{B}

+ 5,{A, !E1, !E2, !E3, !E4, E5}P5{B}

+ 6,{A, !E1, !E2, !E3, !E4, !E5, E6}P6{B}

+ 7,{A, !E1, !E2, !E3, !E4, !E5, !E6, E7}P7{B}

+ 8,{A, !E1, !E2, !E3, !E4, !E5, !E6, !E7}=>L {B}

* Lại có:

+ {A,E8}P8{B}

+ {A,!E8} =>L  {B}