

# Design and Analysis of Algorithms

Tran Ba Quan Thuy

November 2018

Project Reporting

## 1 Introduction

Hiện nay việc sắp xếp container giao cho khách hàng vẫn diễn ra một cách thủ công nên tốn rất nhiều chi phí. Bài toán đặt ra là phải sắp xếp các container sao cho số trên container nhỏ được xếp lên trên số trên container lớn với số bước nhỏ nhất.

## 2 Design and Analysis

### 1. Design and Analysis Algorithms :

- Ý tưởng 1 : sử dụng khi kích cỡ chiều dài và chiều cao của container trên thuyền và bãi giống nhau
  - Object :
    - \* Container : Các thuộc tính gồm số trên container , vị trí của container, ...
    - \* AvailableSlot : Kiểm tra chỗ còn trống gồm các thuộc tính vị trí dòng , cột và giá trị true / false tương ứng của nó.
    - \* SortingObject : Đối tượng sử dụng để sắp xếp.
    - \* InputReader : Đối tượng sử dụng để đọc file text.
  - Algorithm :
    - \* Cho trước một mảng hai chiều input đầu vào. Sau đó truyền mảng input đó vào 1 mảng 2 chiều gồm các container , tạo 1 mảng 2 chiều để lưu kết quả và 1 list để lưu quá trình bốc từng container
    - \* Thuận toán : Tạo 1 mảng 1 chiều AvailableSlot bằng với kích cỡ của số cột input truyền vào. Bóc từng cột của mảng 2 chiều sau đó cho vào 1 list và sắp xếp chọn ra phần tử lớn nhất. Sau đó sẽ dựa trên AvailableSlot và vị trí con trỏ ở mảng kết quả để đưa container được chọn vào vị trí thích hợp.

- \* Ưu điểm : thỏa mãn được ý tưởng ban đầu sẽ sắp xếp được với kích thước bằng kích thước input lúc đưa vào
- \* Nhược điểm : chưa phải là thuật toán có số bước tối ưu nhất có thể, thuật toán cần xử lý nhiều câu lệnh if, sử dụng mảng 2 chiều và mảng 1 chiều nên rất khó để kiểm soát và sửa lỗi.
- Ý tưởng 2 : sử dụng với kích cỡ chiều cao sân bãi cho trước và chiều dài sân không giới hạn
  - Object :
    - \* Container : Các thuộc tính gồm số trên container , vị trí của container, ...
    - \* SortingObject : Đối tượng sử dụng để sắp xếp
  - Algorithm :
    - \* Cho trước một mảng hai chiều input đầu vào. Sau đó truyền mảng list gồm các n stack chứa container với n là số cột của mảng input đầu vào.
    - \* Thuật toán : bóc tất cả các container trong các stack ra và kiểm tra lần lượt, tìm ra container có số lớn nhất trong đó sau đó đẩy qua 1 list gồm các queue container. Nếu queue vượt quá chiều cao sân bãi hoặc giá trị container cần đưa vào lớn hơn container đang chứa trong queue list sẽ tự động sinh ra 1 queue mới và chứa container đó. Sau khi chứa vào 1 list truy vết cách lựa chọn cũng sẽ thêm container được chọn vào
    - \* Ưu điểm : thỏa mãn được yêu cầu với số bước nhỏ nhất và chiều cao sân tương ứng. Thuật toán dùng stack và queue nên rất dễ sửa lỗi và kiểm soát.
    - \* Nhược điểm : không tận dụng được tối đa chiều dài sân
    - \* Điểm mở rộng : có thể tận dụng được tối ưu chiều dài sân bằng cách sử dụng availableslot ở ý tưởng 1 nếu yêu cầu bài toán giới hạn sân
    - \* Mã giả :
      - while list<Stack> != empty
      - for all stack in list<stack>
      - for all element in stack
      - if stack.get(element) != 0
      - add element to tempListSorting
      - end if
      - end for
      - sort tempListSorting
      - add max tempListSorting to ListResult
      - end for
      - end while

## 2. Design GUI and The Output

- Object :
  - ContainerBox : hiển thị container
  - SailBoat : mô phỏng tàu chở hàng đến cảng
  - Rectangle ( implement inside java ) : bãi sân
- Requirement :
  - Vẽ được containerbox, tàu, và bãi.
  - Thể hiện được đường đi của thuật toán.
  - Tìm file trên chính giao diện.
- Extension :
  - Khi thể hiện đường đi thuật toán có thanh hiển thị tiến trình để người sử dụng biết được tiến trình còn bao nhiêu.
  - Chọn màu sắc phù hợp để không làm người sử dụng khó thấy.
  - Có thể dễ dàng nâng cấp, thêm tính năng mới.
  - Khi tìm file có chức năng properties để lưu được vị trí file cần tìm cho người sử dụng.
  - Có hàm main để kiểm thử các test case

### 3. Coding

- Phát triển phần mềm theo hướng đối tượng, chia thành 3 module chính và 1 module phụ.
- Module chính :
  - Entity ( Thực thể ) : Chứa các lớp đối tượng để lưu thông tin trong 1 vòng đời của phần mềm ( Container, AvailableSlot,... )
  - Boundary ( Giao diện ) : là module để hiển thị giao diện cho người dùng
  - Algorithm ( Thuật toán ) : chứa các xử lý giải thuật về sắp xếp thùng hàng
- Module phụ
  - Asset : chứa các hình ảnh, file cấu hình đường dẫn,...

### 4. Kiểm thử

- Requirement analysis - Phân tích yêu cầu
  - Xác định đầu vào của bài toán : file text chứa dữ liệu đầu vào bao gồm chiều cao container trên tàu , mảng 2 chiều container, thông tin chiều cao sân bãi ,...

- Xác định quá trình hoạt động của thuật toán : xác định được quá trình thuật toán chạy sau đó so sánh kết quả của phần mềm với giá trị dự tính của mẫu thử test case và đưa ra kết luận.
- Xác định đầu ra của bài toán : thuật toán trong đề bài là tuyệt đối với 1 test case nhất định( có nghĩa với 1 test case cụ thể thì chỉ cho ra 1 kết quả duy nhất ở tất cả các lần kiểm thử)
- Điểm mở rộng : có thể làm ra 1 phần mềm nhỏ tự sinh ra test case hoặc sử dụng lớp Alogorithm ( giải thuật ) để kiểm tra kết quả đầu ra
- Lựa chọn loại kiểm thử :
  - Unit Test :
    - \* Phát hiện được những điều kiện thiếu sót của thuật toán.
    - \* Tìm lỗi của phần mềm dễ dàng.
    - \* Chi phí thực hiện thấp.
    - \* Giúp phát hiện lỗi nhanh để kịp thời chỉnh sửa
    - \* Nhược điểm: không thể kiểm thử hết tất cả trường hợp với các bài toán khó và phức tạp. Kết quả của các test case chỉ có tính chất tương đối nên phải kiểm thử rất nhiều lần để có được độ tin cậy cao.
  - Integration Test:
    - \* Kiểm thử tính đúng đắn trình tự khi chọn dữ liệu của phần mềm ( phải chọn file input trước khi sắp xếp, và thực hiện hóa bài toán,...)
    - \* Kiểm thử tính nhất quán ( toàn vẹn) của các đối tượng và hàm khi kết hợp với nhau( giữa giao diện với thuật toán , giữa thuật toán với các lớp thực thể như container,...)

### 3 Conclusion and Result

- Ưu điểm :
  - Giải quyết được vấn đề chính của bài toán.
  - Có 1 số tính năng mở rộng và yêu cầu phi chức năng.
- Nhược điểm
  - Không thể ảo hóa với số lượng nhiều container
  - Phần mềm tuy đã đáp ứng được yêu cầu bài toán về sắp xếp nhưng vẫn còn nhiều điểm cần hoàn thiện và nâng cấp.

## 4 Related Document

<https://viblo.asia/p/quy-trinh-kiem-thu-phan-mem-software-testing-life-cycle-stlc-Qbq5QLvmlD8>

<http://asq.org/learn-about-quality/quality-assurance-quality-control/overview/overview.html>

<http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms>