Shape, square

Description automatically generated

**TRƯỜNG ĐẠI HỌC LAO ĐỘNG – XÃ HỘI**

**Ngành: Công nghệ thông tin**

**----------o0o----------**

**Logo

Description automatically generated**

**BÀI BÁO CÁO**

**TRÍ TUỆ NHÂN TẠO**

**Đề tài: THUẬT TOÁN GREEDY**

**Sinh viên thực hiện:** Nguyễn Văn Quân

Vũ Ngọc Phương

Vũ Đức Thanh

Hoàng Quốc Tráng

**Lớp tín chỉ:** D17CN01

**Giảng viên hướng dẫn:** Nguyễn Anh Thơ

**Hà Nội, năm 2023**

# **LỜI MỞ ĐẦU**

Ngày nay, chúng ta đã bước vào thế kỷ 21, kỷ nguyên của Công nghệ thông

tin, đặc biệt là trí tuệ nhân là yếu tố quan trọng nhất quyết định sự thành công của

mỗi ngành hay mỗi quốc gia. Trí tuệ nhân tạo đã và đang làm thay đổi cuộc sống

của chúng ta, với sự phát triển mạnh mẽ của việc áp dụng các nghiên cứu về trí tuệ

nhân tạo áp dụng cho cuộc sống. Tất cả các ngành như: Quân đội, y tế, giáo dục, kinh tế thương mại, tài chính. Đều có thể áp dụng trí tuệ nhân một cách rộng rãi. Việc áp dụng trí tuệ nhân tạo để giải quyết các vấn đề trong xã hội và việc phát triển kinh tế đang được nhà nước khuyến khích và đầu tư rất lớn.

Trên thế giới cũng như Việt Nam, CNTT có ảnh hưởng rất mạnh mẽ đến sự

phát triển của đất nước và thế giới đặc biệt là trí tuệ nhân tạo. Nó chở thành một yếu tố không thể thiếu và có tính quyết định đến sự thành công hay thất bại của nhiều ngành ở nước ta, CNTT đang phát triển với tốc độ khá mạnh mẽ và được ứng dụngrộng rãi trong tất cả các lĩnh vực, đặc biệt là trong công tác ứng dụng công nghệ vào trong cuộc sống.

Như chúng ta đã biết, sức mạnh của một nền kinh tế phụ thuộc rất lớn vào

các hoạt động trong nước của các doanh nghiệp, vì vậy sự thành công trong kinh

doanh của doanh nghiệp không những là mục tiêu của riêng doanh nghiệp, mà nó

còn là nhân tố quyết định vị thế của đất nước trên trường quốc tế. Việc đưa AI vào

áp dụng cho các doanh nghiệp và cả các ngành như y tế, công nghiệp nặng đang

được ưu tiên và phát triển mạnh mẽ.

Tại Việt Nam, Nhà nước đang đi vào phát triển dịch vụ, và đầu tư mạnh mẽ

vào trí tuệ nhân tạo hay còn gọi là AI. Vì thế, việc đó đầu nó và phát triển nó đang là một xu thế rất hot và rất được ưu chuộng hiện này.

Chính vì vậy thông qua việc học môn trí tuệ nhân tạo (AI) nhóm em đã nghĩ

ra một ý tưởng nhỏ đó là áp dụng thuật toán mình đã học để làm ra một Game xếp

hình 8 ô số. Phục vụ cho việc chứng minh áp dụng trí tuệ nhân tạo mang lại lợi ích

tối ưu về không gian và thời gian cho con người.

# **LỜI CẢM ƠN**

Trong thời đại công nghệ 4.0 đang ngày càng phát triển, ứng dụng trí tuệ nhân tạo vào đời sống được xem là một trong những xu thế phát triển mạnh mẽ hiện nay. Việc tự động hóa các hành vi thông minh đang trở thành một cuộc cách mạngtrong ngành CNTT.

Vì vậy chúng em đã thực hiện đồ án trí tuệ nhân tạo áp dụng thuật toán tham lam để làm ra bài toán sắp xếp công việc, phục vụ cho việc chứng minh áp dụng trí tuệ nhân tạo mang lại lợi ích tối ưu về không gian và thời gian cho con người.

Với sự hướng dẫn, động viên tận tình của thầy Nguyễn Anh Thơ, chúng em đã hiểu được cách thức hoạt động của thuật toán và hoàn thành bài báo cáo đồ án này. Do chưa có nhiều kinh nghiệm nghiên cứu, thực hành nên chúng em cũng không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và góp ý của thầy để đề tài của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

[**LỜI MỞ ĐẦU** 2](#_Toc153573936)

[**LỜI CẢM ƠN** 3](#_Toc153573937)

[**CHƯƠNG I: GIỚI THIỆU TRÍ TUỆ NHÂN TẠO** 5](#_Toc153573938)

[**1. Khái niệm Trí Tuệ Nhân Tạo.** 5](#_Toc153573939)

[**2. Đặc trưng của trí thông minh.** 5](#_Toc153573940)

[*Hình 1: Các đặc trưng của AI.* 5](#_Toc153573941)

[**2.1. Học và Tự Đào Tạo (Learning and Self-training):**. 6](#_Toc153573942)

[**2.2. Tính Nhận Thức (Perception):** 6](#_Toc153573943)

[**2.3. Khả Năng Giải Quyết Vấn Đề (Problem Solving):** 6](#_Toc153573944)

[**2.4. Ngôn Ngữ Tự Nhiên (Natural Language Processing - NLP):** 6](#_Toc153573945)

[**2.5. Tính Linh Hoạt (Adaptability):** 6](#_Toc153573946)

[**3. Các thành phần chính của AI.** 7](#_Toc153573947)

[*Hình 2: Các thành phần của AI.* 7](#_Toc153573948)

[**3.1. Máy học (Machine Learning):** 7](#_Toc153573949)

[**3.2. Mạng nơ-ron (Neural Networks):** 8](#_Toc153573950)

[**3.3. Thuật toán AI:** 8](#_Toc153573951)

[**3.4. Dữ liệu:** 8](#_Toc153573952)

[**3.5. Giao diện Người dùng (User Interface):** 8](#_Toc153573953)

[**3.6. Quyết định và Lập lịch (Decision and Planning):** 9](#_Toc153573954)

[**3.7. Hệ thống Tương Tác (Interaction Systems):** 9](#_Toc153573955)

[**3.8. Quản lý Dữ liệu và Lưu trữ (Data Management and Storage):** 9](#_Toc153573956)

[**3.9. Tăng cường và Bảo mật (Enhancement and Security):** 9](#_Toc153573957)

[**4. Mục tiêu nghiên cứu của đề tài.** 9](#_Toc153573958)

[**4.1. Nắm vững Lý Thuyết Thuật Toán Tham Lam.** 9](#_Toc153573959)

[**4.2. Áp Dụng Thuật Toán Tham Lam Trong Các Bài Toán Cụ Thể.** 9](#_Toc153573960)

[**4.3. Mở Rộng Ứng Dụng trong Mô Hình Học Máy.** 10](#_Toc153573961)

[**4.4. Đề Xuất Hướng Nghiên Cứu Tiếp Theo.** 10](#_Toc153573962)

[**4.5. Tổng Kết và Đánh Giá.** 10](#_Toc153573963)

[**CHƯƠNG II: THUẬT TOÁN GREEDY** 11](#_Toc153573964)

[**1. Lý thuyết cơ bản về Thuật Toán Tham Lam.** 11](#_Toc153573965)

[**2. Thuật toán.** 11](#_Toc153573966)

[**3. Phương pháp sử dụng thuật toán Greeady.** 12](#_Toc153573967)

[**CHƯƠNG III: BÀI TOÁN ỨNG DỤNG GREEDY** 13](#_Toc153573968)

[**1. Bài toán ứng dụng Sắp xếp công việc.** 13](#_Toc153573969)

[**2. Bài toán ứng dụng Mua sách.** 17](#_Toc153573970)

# 

# **CHƯƠNG I: GIỚI THIỆU TRÍ TUỆ NHÂN TẠO**

## **1. Khái niệm Trí Tuệ Nhân Tạo.**

• Trí tuệ nhân tạo (AI) là lĩnh vực nghiên cứu làm cho máy tính có thể làm được các việc như có trí thông minh của con người.

## **2. Đặc trưng của trí thông minh.**

****

*Hình 1: Các đặc trưng của AI.*

**Trí thông minh nhân tạo (AI) có những đặc trưng quan trọng sau đây:**

**2.1. Học và Tự Đào Tạo (Learning and Self-training):** Một trong những đặc điểm chính của AI là khả năng học từ dữ liệu và trải nghiệm. Hệ thống AI có thể tự đào tạo thông qua thuật toán máy học và điều chỉnh hiệu suất của mình theo thời gian.

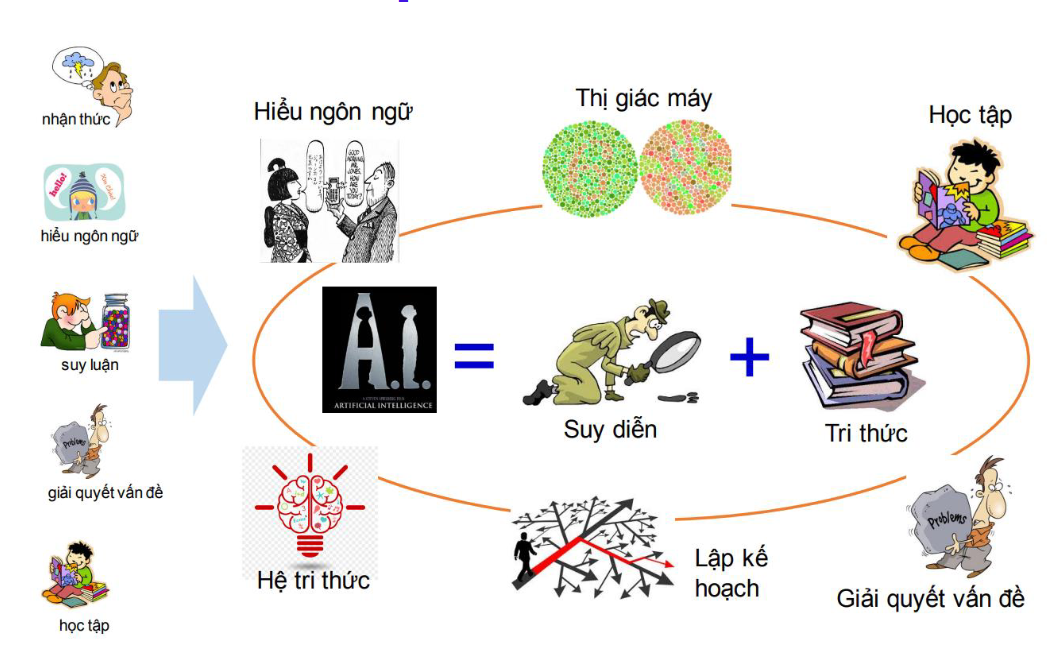
**2.2. Tính Nhận Thức (Perception):** Một số hệ thống AI có khả năng nhận thức và hiểu các thông tin từ thế giới bên ngoài thông qua cảm biến và dữ liệu đầu vào.

**2.3. Khả Năng Giải Quyết Vấn Đề (Problem Solving):** AI được thiết kế để giải quyết các vấn đề phức tạp và thậm chí là tìm ra giải pháp mà con người có thể chưa nghĩ đến.

## **2.4. Ngôn Ngữ Tự Nhiên (Natural Language Processing - NLP):** AI có khả năng hiểu và tương tác bằng ngôn ngữ tự nhiên, giúp người dùng giao tiếp một cách tự nhiên với hệ thống.

## **2.5. Tính Linh Hoạt (Adaptability):** AI có khả năng thích ứng với môi trường mới, tình huống mới, và dữ liệu mới mà nó chưa từng trải qua. Điều này làm cho hệ thống AI trở nên linh hoạt trong nhiều ngữ cảnh.

## **3. Các thành phần chính của AI.**

****

*Hình 2: Các thành phần của AI.*

Trong lĩnh vực Trí tuệ nhân tạo (AI), có nhiều thành phần chính đóng vai trò quan trọng trong quá trình phát triển và triển khai các hệ thống AI. Dưới đây là một số thành phần quan trọng:

## **3.1. Máy học (Machine Learning):**

* Học giám sát (Supervised Learning): Mô hình được đào tạo bằng cách sử dụng dữ liệu có nhãn.
* Học không giám sát (Unsupervised Learning): Mô hình học từ dữ liệu không có nhãn.
* Học tăng cường (Reinforcement Learning): Mô hình học từ trải nghiệm và phản hồi từ môi trường.

## **3.2. Mạng nơ-ron (Neural Networks):**

* Mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN): Được lấy cảm hứng từ cách hoạt động của não người.
* Mạng nơ-ron hồi quy (Recurrent Neural Networks - RNN): Có khả năng xử lý dữ liệu tuần tự và chuỗi.
* Mạng nơ-ron chập (Convolutional Neural Networks - CNN): Thích hợp cho xử lý ảnh và video.

## **3.3. Thuật toán AI:**

* Thuật toán học máy: Bao gồm các thuật toán như Support Vector Machines, Decision Trees, Random Forests, và k-Nearest Neighbors.
* Thuật toán học sâu (Deep Learning Algorithms): Được sử dụng trong các mô hình sâu và phức tạp như mạng nơ-ron sâu.

## **3.4. Dữ liệu:**

* Dữ liệu đào tạo (Training Data): Dữ liệu được sử dụng để đào tạo mô hình máy học.
* Dữ liệu thử nghiệm (Testing Data): Dữ liệu được sử dụng để đánh giá hiệu suất của mô hình.
* Dữ liệu thực tế (Real-world Data): Dữ liệu mà mô hình sẽ phải xử lý khi triển khai.

## **3.5. Giao diện Người dùng (User Interface):**

* Giao diện Người dùng Dựa trên Văn bản (Natural Language Processing - NLP): Cho phép tương tác bằng ngôn ngữ tự nhiên.
* Giao diện đồ họa: Cho phép người dùng tương tác với và kiểm soát hệ thống AI thông qua giao diện đồ họa.

## **3.6. Quyết định và Lập lịch (Decision and Planning):**

* Hệ thống quyết định thông minh: Các thành phần có khả năng đưa ra quyết định dựa trên dữ liệu và học từ trải nghiệm.

## **3.7. Hệ thống Tương Tác (Interaction Systems):**

* Hệ thống Tương tác Người-Máy: Các công nghệ cho phép tương tác trực tiếp và linh hoạt giữa người và máy.

## **3.8. Quản lý Dữ liệu và Lưu trữ (Data Management and Storage):**

* Cơ sở dữ liệu: Lưu trữ và quản lý dữ liệu cần thiết cho quá trình đào tạo và triển khai.

## **3.9. Tăng cường và Bảo mật (Enhancement and Security):**

* Tăng cường hiệu suất: Các phương pháp và công nghệ để tăng cường hiệu suất của mô hình và hệ thống AI.
* Bảo mật AI: Bảo vệ mô hình và dữ liệu khỏi rủi ro an ninh.

## **4. Mục tiêu nghiên cứu của đề tài.**

**4.1. Nắm vững Lý Thuyết Thuật Toán Tham Lam.**

* Hiểu rõ về nguyên lý hoạt động của thuật toán tham lam và cách nó tối ưu hóa giải pháp thông qua việc chọn lựa các quyết định cục bộ.
* So sánh và phân tích sự khác biệt giữa thuật toán tham lam và các phương pháp tối ưu hóa khác, như quy hoạch động và tìm kiếm.

**4.2. Áp Dụng Thuật Toán Tham Lam Trong Các Bài Toán Cụ Thể.**

* Nghiên cứu và thực hiện áp dụng thuật toán tham lam trong các bài toán cụ thể như tìm đường đi ngắn nhất, bài toán bao phủ tối thiểu, và lập lịch công việc.
* Phân tích kết quả và so sánh hiệu suất của thuật toán tham lam với các thuật toán khác trong các tình huống thực tế.

**4.3. Mở Rộng Ứng Dụng trong Mô Hình Học Máy.**

* Nghiên cứu cách thuật toán tham lam có thể được tích hợp vào mô hình học máy, đặc biệt trong việc chọn lựa các biến quan trọng hoặc tối ưu hóa hàm mất mát.
* Đánh giá ứng dụng và tiềm năng của thuật toán tham lam trong môi trường học máy.

**4.4. Đề Xuất Hướng Nghiên Cứu Tiếp Theo.**

* Xác định những thách thức và hạn chế của thuật toán tham lam trong các bài toán cụ thể.
* Đề xuất hướng nghiên cứu và phát triển tiếp theo để cải thiện hiệu suất và ứng dụng của thuật toán tham lam.

**4.5. Tổng Kết và Đánh Giá.**

* Tổng hợp kiến thức thu được từ nghiên cứu và áp dụng thuật toán tham lam trong nhiều bối cảnh.
* Đánh giá ảnh hưởng và giá trị của thuật toán tham lam trong việc giải quyết các vấn đề thực tế.

Mục tiêu của nghiên cứu này là xây dựng một hiểu biết sâu rộng về thuật toán tham lam và khám phá khả năng ứng dụng của nó trong nhiều lĩnh vực khác nhau, từ các vấn đề cơ bản đến các ứng dụng tiên tiến trong môi trường học máy.

# **CHƯƠNG II: THUẬT TOÁN GREEDY**

## 

## **1. Lý thuyết cơ bản về Thuật Toán Tham Lam.**

Thuật toán tham lam (Greedy Algorithm) là một phương pháp giải quyết vấn đề tìm kiếm cực tiểu hoặc cực đại thông qua quyết định tại từng bước một, mà không cần xem xét lại quyết định đã đưa ra. Phương pháp này chọn lựa tại mỗi bước để đạt được giải pháp tối ưu toàn cục.

Ý tưởng chính:  
 - Lựa chọn tối ưu cục bộ tại mỗi bước  
 - Mong muốn tìm ra lựa chọn tối ưu toàn cục.  
 - Giải thuật tham lam thường không mang tính tổng quát.

- Cần có điểm tựa vững chắc về mặt toán học.  
 - Độ phức tạp thời gian thường tốt hơn nhiều so với  
 - Duyệt toàn bộ và Nhánh cận

## **2. Thuật toán.**

Nói chung, giải thuật tham lam có năm thành phần:

* Một tập hợp các ứng viên (candidate), để từ đó tạo ra lời giải
* Một hàm lựa chọn, để theo đó lựa chọn ứng viên tốt nhất để bổ sung vào lời giải
* Một hàm khả thi (feasibility), dùng để quyết định nếu một ứng viên có thể được dùng để xây dựng lời giải
* Một hàm mục tiêu, ấn định giá trị của lời giải hoặc một lời giải chưa hoàn chỉnh
* Một hàm đánh giá, chỉ ra khi nào ta tìm ra một lời giải hoàn chỉnh.

Có hai thành phần quyết định nhất tới quyết định tham lam:

***Tính chất lựa chọn tham lam***

Chúng ta có thể lựa chọn giải pháp nào được cho là tốt nhất ở thời điểm hiện tại và sau đó giải bài toán con nảy sinh từ việc thực hiện lựa chọn vừa rồi. Lựa chọn của thuật toán tham lam có thể phụ thuộc vào các lựa chọn trước đó. Nhưng nó không thể phụ thuộc vào một lựa chọn nào trong tương lai hay phụ thuộc vào lời giải của các bài toán con. Thuật toán tiến triển theo kiểu thực hiện các chọn lựa theo một vòng lặp, cùng lúc đó thu nhỏ bài toán đã cho về một bài toán con nhỏ hơn. Đấy là khác biệt giữa thuật toán này và giải thuật Quy Hoạnh Động. Giải thuật quy hoạch động duyệt hết và luôn đảm bảo tìm thấy lời giải. Tại mỗi bước của thuật toán, quy hoạch động đưa ra quyết định dựa trên các quyết định của bước trước, và có thể xét lại đường đi của bước trước hướng tới lời giải. Giải thuật tham lam quyết định sớm và thay đổi đường đi thuật toán theo quyết định đó, và không bao giờ xét lại các quyết định cũ. Đối với một số bài toán, đây có thể là một thuật toán không chính xác.

***Cấu trúc con tối ưu***

Một bài toán được gọi là “có cấu trúc tối ưu”, nếu một lời giải tối ưu của bài toán con chứa lời giải tối ưu của bài toán lớn hơn.

## **3. Phương pháp sử dụng thuật toán Greeady.**

* **Xác định mục tiêu tối ưu:** Xác định rõ mục tiêu cần đạt được và hàm mục tiêu cần được tối ưu hóa.
* **Xác định các bước lựa chọn:** Chia bài toán thành các bước nhỏ hơn và xác định cách lựa chọn tại mỗi bước để tiến gần hơn đến mục tiêu tối ưu.
* **Xác định hàm đánh giá:** Đưa ra một hàm đánh giá cho mỗi lựa chọn tại mỗi bước, đánh giá xem nó đưa lại độ tốt nào đối với mục tiêu tối ưu.
* **Sắp xếp các lựa chọn:** Sắp xếp các lựa chọn theo thứ tự giảm dần hoặc tăng dần của hàm đánh giá.
* **Chọn lựa tốt nhất:** Chọn lựa chọn tốt nhất tại mỗi bước và cập nhật trạng thái của bài toán.
* **Lặp lại cho đến khi đạt được mục tiêu hoặc không thể tiếp tục:** Lặp lại các bước trên cho đến khi đạt được mục tiêu hoặc không thể tiếp tục nữa.

Ví dụ: Bây giờ mẹ bạn cho bạn 2 tờ tiền mệnh giá 100.000 đ và 200.000 đ và bạn chỉ được chọn 1. Và đương nhiên mình sẽ chọn tờ 200.000 đ vì nó giá trị hơn mặc dù số lượng và kích thước của 2 tờ đều như nhau.

Một ví dụ khác nhé . Ta có một ba lô có trọng lượng là 37 và 4 loại đồ vật với trọng lượng và giá trị tương ứng, yêu cầu ở đây là bạn sẽ phải chọn tối đa số lượng đồ vật để vừa phù hợp với trọng lượng của ba lô mà giá trị lấy được là lớn nhất.

Từ đó ta có kỹ thuật Tham lam áp dụng cho bài toán này là:

1. Tính đơn giá cho các loại đồ vật.
2. Xét các loại đồ vật theo thứ tự đơn giá từ lớn đến nhỏ.
3. Với mỗi đồ vật được xét sẽ lấy một số lượng tối đa mà trọng lượng còn lại của ba lô cho phép.
4. Xác định trọng luợng còn lại của ba lô và quay lại bước 3 cho đến khi không còn có thể chọn được đồ vật nào nữa.

Demo tý nhé : Ta có một ba lô có trọng lượng là 37 và 4 loại đồ vật với trọng lượng và giá trị tương ứng được cho như sau :

Loại đồ vật : A - B - C - D

Trọng lượng : 15 - 10 - 2 - 4

Giá trị : 30 - 25 - 2 - 6

Từ bảng đã cho ta tính đơn giá cho các loại đồ vật và sắp xếp các loại đồ vật này theo thứ tự đơn giá giảm dần ta có bảng sau.

Loại đồ vật : B - A - D - C

Trọng lượng : 10 - 15 - 4 - 2

Giá trị : 25 - 30 - 6 - 2

Đơn giá : 2.5 - 2.0 - 1.5 - 1.0

Theo đó thì thứ tự ưu tiên để chọn đồ vật là là B, A, D và cuối cùng là C.

Vật B được xét đầu tiên và ta chọn tối đa 3 cái vì mỗi cái vì trọng lượng mỗi cái là 10 và ba lô có trọng lượng 37. Sau khi đã chọn 3 vât loại B, trọng lượng còn lại trong ba lô là 37 – 3\*10 = 7. Ta xét đến vật A, vì A có trọng lượng 15 mà trọng lượng còn lại của balô chỉ còn 7 nên không thể chọn vật A. Xét vật D và ta thấy có thể chọn 1 vật D, khi đó trọng lượng còn lại của ba lô là 7-4 = 3. Cuối cùng ta chọn được một vật C.

Như vậy chúng ta đã chọn 3 cái loại B, một cái loại D và 1 cái loại C. Tổng trọng lương là 3*10 + 1*4 + 1*2 = 36 và tổng giá trị là 3*25+1*6+1*2 = 83.

# **CHƯƠNG III: BÀI TOÁN ỨNG DỤNG GREEDY**

## **1. Bài toán ứng dụng Sắp xếp công việc.**

**Bài toán:**  
• Cho tập gồm n công việc, mỗi công việc được biểu diễn bởi cặp thời gian bắt đầu si và thời ­­-gian kết thúc fi (i=1, 2, ..., n).  
• Hãy lựa chọn nhiều nhất các công việc có thể thực hiện tuần tự bởi một máy hoặc một cá nhân mà không xảy ra tranh chấp.  
• Mỗi công việc chỉ thực hiện đơn lẻ tại một thời điểm

Ví dụ: Input:  
• Số lượng công việc: 6  
•Thời gian bắt đầu Start [= { 1, 3, 0, 5, 8, 5}  
•Thời gian kết thúc Finish[]= { 2, 4, 6, 7, 9, 9}

Output:  
•OPT[] = {0, 1, 3, 4}

**Giải thuật**

Input:  
Sắp xếp công việc ( N,S[], F[]):  
• N là số lượng công việc.  
S[] thời gian bắt đầu.  
F[] thời gian kết thúc.  
Ouput: Danh sách thực thi nhiều nhất.  
Actions:  
• Bước 1 Sắp xếp thứ tự tăng dần của thời gian  
kết thúc.  
• Bước 2 (Khởi tạo) Lựa chọn công việc đầu tiên  
làm phương án tối ưu ( OPT=1). N = = N\{1};  
• Bước 3 (Lặp).  
Với mỗi công việc j eN  
if (S[j] >=F[i]) {OPT = OPTUj; i = j; N = N\{i} }  
• Bước 4 (Trả lại kết quả)

**Cài đặt chương trình**

#include <iostream>

#include <fstream>

#include <algorithm>

#include <iomanip>

using namespace std;

const int MAX = 6;

int s[MAX] ;

int f[MAX] ;

int a[MAX];

bool sosanh (int i, int j) {

return f[i] < f[j];

}

void printHorizontalLine(int width) {

cout << "\t\t\t\t\t\t";

for (int i = 0; i < width; i++) {

cout << "-";

}

cout << endl;

}

void printTableHeader() {

printHorizontalLine(49); // Total width is 50 characters

cout << "\t\t\t\t\t\t| " << setw(13) << "Cong Viec" << " | " << setw(13) << "Time Bat Dau" << " | " << setw(13) << "Time Ket Thuc" << " |" << endl;

printHorizontalLine(49);

}

void printTableRow(int index, int startTime, int endTime) {

cout << "\t\t\t\t\t\t| " << setw(13) << (index + 1) << " | " << setw(13) << startTime << " | " << setw(13) << endTime << " |" << endl;

}

int main() {

ifstream file("danhsachcongviec.txt");

if (!file) {

cerr << "Khong the mo file danhsachcongviec.txt.\n";

return 1;

}

cout << "Dang doc file danhsachcongviec.txt\n";

int n;

file >> n;

if (n != MAX) {

cerr << "So luong cong viec khong hop le.\n";

return 1;

}

for (int i = 0; i < n; i++) {

file >> s[i];

}

for (int i = 0; i < n; i++) {

file >> f[i];

}

file.close();

cout << "\n\n\t\t\t\t\t\t\tDANH SACH CONG VIEC BAN DAU\n";

printTableHeader();

for (int i = 0; i < MAX; i++) {

printTableRow(i, s[i], f[i]);

}

printHorizontalLine(49);

cout << "\n\n";

cout << "\t\t\t\t\t\t\tDANH SACH CONG VIEC DUOC SAP XEP\n";

printTableHeader();

// Chèn và in thông tin cho công việc đầu tiên

for (int i = 0; i < MAX; i++) a[i] = i;

sort(a, a+MAX, sosanh);

a[0] = 0;

cout << "\t\t\t\t\t\t| " << setw(13) << "1" << " | " << setw(13) << s[a[0]] << " | " << setw(13) << f[a[0]] << " |" << endl;

int e = f[a[0]];

// Chèn và in thông tin cho các công việc tiếp theo

for (int i = 1; i < MAX; i++)

if (e <= s[a[i]]) {

cout << "\t\t\t\t\t\t| " << setw(13) << (i + 1) << " | " << setw(13) << s[a[i]] << " | " << setw(13) << f[a[i]] << " |" << endl;

e = f[a[i]];

}

printHorizontalLine(49);

}

## **2. Bài toán ứng dụng Mua sách.**

Trong nhà sách đang bán một bộ sách gồm n cuốn sách mỗi cuốn sách có giá là a[i]. Nhà sách đang có ưu đãi là mua 3 cuốn sách thì miễn phí cho quyển sách rẻ nhất trong 3 cuốn đấy. Tìm chi phí thấp nhất để mua hết n cuốc sách.

Ví dụ:  
7  
5 3 6 9 1 2 2  
Đáp án: (9+6+5) + (3+2+2) +1 = 21  
21

Input:  
• Số lượng giá tiền mua sách: 7  
•Giá tiền bắt đầu Start [= { 5, 3, 6, 9, 1, 2, 2}  
•Giá tiền kết thúc Finish[]= { 9, 6, 3, 2, 1}

Output:  
•OPT[] = {5, 2}

**Sắp xếp và xử lý logic**

Theo thứ tự từ lớn đến nhỏ

9 6 5 3 2 2 1

Xử lý logic nhóm lần lượt giá của ba cuốn sách có giá tiền lớn nhất lần lượt cho đến hết là sẽ được chi phí thấp nhất.

**Giải thuật**

Input:  
Sắp xếp thứ tự giá sách ( N,S[], F[]):  
• N là số lượng giá.  
S[] Giá tiền bắt đầu.  
F[] Giá tiền kết thúc.  
Ouput: Danh sách thực thi nhiều nhất.  
Actions:  
• Bước 1 Sắp xếp thứ tự giảm dần của giá tiền  
kết thúc.  
• Bước 2 (Khởi tạo) Lựa chọn những giá tiền nhỏ nhất   
làm phương án tối ưu ( OPT=5, 2). N = = N\{1};  
• Bước 3 (Lặp).  
Với mỗi giá tiền j eN  
if (S[j] <=F[i]) {OPT = OPTUj; i = j; N = N\{i} }  
• Bước 4 (Trả lại kết quả)

**Cài đặt chương trình**

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

#include <iomanip>

using namespace std;

const int MAX\_SIZE = 1005;

int n;

double ans = 0;

struct Book {

double price;

int originalIndex;

};

vector<Book> books;

vector<int> sortedIndices;

// Hàm dọc dữ liệu từ file

void readFile(const char\* filename) {

ifstream inputFile(filename);

if (!inputFile) {

cerr << "Khong the mo file " << filename << endl;

exit(EXIT\_FAILURE);

}

inputFile >> n;

books.resize(n + 1);

sortedIndices.resize(n + 1);

for (int i = 1; i <= n; i++) {

inputFile >> books[i].price;

books[i].originalIndex = i;

sortedIndices[i] = i;

}

inputFile.close();

}

// Hàm so sánh đã sắp xếp theo giảm dần của giá tiền

bool compareBooks(int i, int j) {

return books[i].price > books[j].price;

}

// In ra một dòng của bảng với khung viền

void printTableRow(int stt, double giaTien) {

cout << "\t\t\t\t\t";

cout << setfill(' ') << setw(2) << left << "|";

cout << setw(5) << left << stt << "| " ;

cout << setw(15) << left << giaTien << "|";

cout << endl;

}

int main() {

const char\* filename = "books.txt";

readFile(filename);

cout << "\n\n";

// In ra thông tin sách truớc khi sắp xếp

cout << "\t\t\t\t Thong tin sach truoc khi sap xep \n" << endl;

cout << "\t\t\t\t\t" << setfill('-') << setw(25) << "-" << setfill(' ') << endl;

cout << "\t\t\t\t\t| STT | Gia tien |" << endl;

cout << "\t\t\t\t\t" << setfill('-') << setw(25) << "-" << setfill(' ') << endl;

for (int i = 1; i <= n; i++) {

printTableRow(books[i].originalIndex, books[i].price);

}

cout << "\t\t\t\t\t" << setfill('-') << setw(25) << "-" << setfill(' ') << endl;

// Sắp xếp cho STT theo giảm dần của giá tiền

sort(sortedIndices.begin() + 1, sortedIndices.end(), compareBooks);

// In ra thông tin sách sau khi sắp xếp

cout << "\n\n";

cout << "\t\t\t\t Thong tin sach sau khi sap xep \n" << endl;

cout << "\t\t\t\t\t" << setfill('-') << setw(25) << "-" << setfill(' ') << endl;

cout << "\t\t\t\t\t| STT | Gia tien |" << endl;

cout << "\t\t\t\t\t" << setfill('-') << setw(25) << "-" << setfill(' ') << endl;

for (int i = 1; i <= n; i++) {

int index = sortedIndices[i];

printTableRow(books[index].originalIndex, books[index].price);

}

cout << "\t\t\t\t\t" << setfill('-') << setw(25) << "-" << setfill(' ') << endl;

// Tính số tiền theo logic

for (int i = 1; i <= n; i++) {

if (i % 3 == 0) {

continue;

}

ans += books[sortedIndices[i]].price;

}

cout << "\n\n\n";

cout << setw(39) << left << "\t\t\t\tCac quyen sach duoc tinh vao tong gia tien:" << endl;

cout << "\t\t\t\t\t" << setfill('-') << setw(25) << "-" << setfill(' ') << endl;

cout << "\t\t\t\t\t| STT | Gia tien |" << endl;

cout << "\t\t\t\t\t" << setfill('-') << setw(25) << "-" << setfill(' ') << endl;

for (int i = 1; i <= n; i++) {

if (i % 3 == 0) {

continue;

}

int index = sortedIndices[i];

cout << "\t\t\t\t\t" << "| " << setw(4) << left << books[index].originalIndex << " | " << setw(14) << left << books[index].price << " |" << endl;

}

cout << "\t\t\t\t\t" << setfill('-') << setw(25) << "-" << setfill(' ') << endl;

cout << "\n" << setw(39) << left << "\t\tGia tien sau khi ap dung logic: " << ans << endl;

return 0;

}