**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC TÂY ĐÔ**

**KHOA KỸ THUẬT – CÔNG NGHỆ**

A logo with a green and pink leaf

Description automatically generated

**Trí tuệ - Sáng tạo - Năng động – Đổi mới**

**NIÊN LUẬN 1**

**BÀI TOÁN CÁI BALO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**SVTH: Nguyễn Chí Nguyện – MSSV: 227060172**

**Cần Thơ – tháng 12 năm 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC TÂY ĐÔ**

**KHOA KỸ THUẬT – CÔNG NGHỆ**

A logo with a green and pink leaf

Description automatically generated

**Trí tuệ - Sáng tạo - Năng động – Đổi mới**

**NIÊN LUẬN 1**

**BÀI TOÁN CÁI BALO**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**SVTH: Nguyễn Chí Nguyện – MSSV: 227060172**

**GVHD: Nguyễn Chí Cường**

**Cần Thơ – tháng 12 năm 2024**

# LỜI CẢM ƠN

Để hoàn thành bài niên luận này, chúng em xin gửi lời tri ân sâu sắc đến:

Ban Giám Hiệu Trường Đại học Tây Đô đã tạo điều kiện thuận lợi về cơ sở vật chất, đặc biệt là hệ thống thư viện hiện đại với nguồn tài liệu phong phú, hỗ trợ chúng em trong quá trình nghiên cứu và học tập.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn chân thành đến Thạc sĩ Nguyễn Chí Cường, người đã tận tình hướng dẫn, giảng dạy và truyền đạt kiến thức một cách chi tiết, giúp chúng em có nền tảng vững chắc để thực hiện bài niên luận này.

Vì kinh nghiệm còn hạn chế và hiểu biết chưa sâu rộng, bài niên luận không tránh khỏi những thiếu sót. Chúng em rất mong nhận được ý kiến đóng góp quý báu từ thầy để hoàn thiện bài làm hơn nữa.

Cuối cùng, chúng em kính chúc thầy dồi dào sức khỏe, luôn thành công và hạnh phúc trong cuộc sống.

**SINH VIÊN THỰC HIỆN**

**Nguyễn Chí Nguyện**

# LỜI CAM ĐOAN

Tôi xin cam đoan rằng Niên luận 1 này là do chính chúng tôi thực hiện, không sao chép dưới bất kỳ hình thức nào hay thuê hoặc nhờ người khác thực hiện.

Dữ liệu và kết quả phân tích trong Niên luận 1 đảm bảo tính chính xác, khách quan và trung thực, không có bất kỳ sự ngụy tạo và điều chỉnh kết quả nghiên cứu bằng sự chủ quan của tác giả.

*Cần Thơ, ngày tháng 12 năm 2024*

**Nguyễn Chí Nguyện**

# TÓM TẮT

Bài toán cái balo là một vấn đề nổi tiếng trong lĩnh vực toán học và khoa học máy tính, thường được sử dụng để giải quyết các bài toán tối ưu hóa phức tạp. Bài toán được mô tả như sau: một người có một chiếc balo với dung tích giới hạn, cần chọn một số đồ vật để mang theo. Mỗi đồ vật có trọng lượng và giá trị riêng, và mục tiêu là làm sao để tổng giá trị của các đồ vật trong balo là lớn nhất, đồng thời đảm bảo tổng trọng lượng của chúng không vượt quá dung tích balo.

Việc giải quyết bài toán cái balo đòi hỏi các phương pháp tiếp cận khác nhau, từ các thuật toán tham lam (greedy) đơn giản, đến các phương pháp động (dynamic programming) để đảm bảo tìm được lời giải tối ưu.

Bài toán cái balo không chỉ là một thách thức kỹ thuật mà còn là cơ hội để áp dụng kiến thức về toán học, logic và lập trình. Thông qua việc tìm giải pháp cho bài toán này, chúng ta có thể học được cách tối ưu hóa tài nguyên, đưa ra các quyết định hợp lý và hiệu quả trong nhiều tình huống khác nhau của cuộc sống và công việc.

# ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN

**NIÊN LUẬN, TIỂU LUẬN, KHOÁ LUẬN**

(Học kỳ: I, Năm 3)

**TÊN ĐỀ TÀI: BÀI TOÁN CÁI BALO**

**GIÁO VIÊN HƯỚNG DẪN:**

|  |  |  |
| --- | --- | --- |
| STT | HỌ VÀ TÊN | MSCB |
| 1 | Nguyễn Chí Cường |  |

**SINH VIÊN THỰC HIỆN:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | HỌ VÀ TÊN | MSSV | THƯỞNG  *(Tối đa 1,0 điểm)* | ĐIỂM |
| 1 | Nguyễn Chí Nguyện | 227060172 |  |  |

|  |  |
| --- | --- |
| **I. HÌNH THỨC** *(Tối đa 1,5 điểm)* |  |
| **II. NỘI DUNG** *(Tối đa 5 điểm)* |  |
| **Ứng dụng** *(tối đa 3 điểm)* |  |
| * Giới thiệu chương trình (0,5 điểm) |  |
| **III. CHƯƠNG TRÌNH DEMO** *(Tối đa 3.5 điểm)* |  |
| **Giao diện thân thiện với người dùng** *(0.5 điểm)* |  |
| **Hướng dẫn sử dụng** *(0.5 điểm)* |  |
| **Kết quả thực hiện đúng với kết quả của phần ứng dụng**   * Giải thuật đúng, thực thi chính xác *(2.5 điểm)* |  |

**Ghi chú:**

1. *Điểm trong khung “sinh viên thực hiện” là điểm kết quả cuối cùng của từng sinh viên trong quá trình thực hiện* Niên luận…/tiểu luận/khoá luận*.*
2. *Nếu sinh viên demo chương trình và trả lời vấn đáp không đạt yêu cầu của giáo viên hướng dẫn thì sinh viên sẽ nhận điểm F cho học phần này.*

Cần Thơ, ngày .......... tháng năm 2024

**GIÁO VIÊN CHẤM**

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

🖎🕮✍

.....................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

|  |
| --- |
| *Cần Thơ, ngày tháng năm 2024* |
| **Giáo viên hướng dẫn** |
| **………………..** |

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc184978312)

[LỜI CAM ĐOAN ii](#_Toc184978313)

[TÓM TẮT iii](#_Toc184978314)

[ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN iv](#_Toc184978315)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN v](#_Toc184978316)

[MỤC LỤC vi](#_Toc184978317)

[MỤC LỤC BẢNG viii](#_Toc184978318)

[MỤC LỤC HÌNH ẢNH ix](#_Toc184978319)

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc184978320)

[1.1 LỜI MỞ ĐẦU 1](#_Toc184978321)

[1.2 MÔ TẢ BÀI TOÁN 1](#_Toc184978322)

[1.3 MỤC TIÊU CẦN ĐẠT ĐƯỢC 2](#_Toc184978323)

[1.4 HƯỚNG GIẢI QUYẾT 2](#_Toc184978324)

[1.4.1 Giải thuật quy hoạch động 2](#_Toc184978325)

[1.4.2 Giải thuật tham lam 2](#_Toc184978326)

[1.5 PHẠM VỊ NGHIÊN CỨU 3](#_Toc184978327)

[1.5.1 Ứng dụng thực tế 3](#_Toc184978328)

[1.5.2 Môi trường triển khai 3](#_Toc184978329)

[1.5.3 Phạm vi dữ liệu dữ liệu 3](#_Toc184978330)

[1.6 KẾ HOẠCH THỰC HIỆN 3](#_Toc184978331)

[CHƯƠNG II: CỞ SỞ LÝ THUYẾT 5](#_Toc184978332)

[2.1 HTML 5](#_Toc184978333)

[2.2 CSS 5](#_Toc184978334)

[2.3 JAVASCRIPT 5](#_Toc184978335)

[2.4 THUẬT TOÁN QUY HOẠCH ĐỘNG 6](#_Toc184978336)

[2.5 THUẬT TOÁN THAM LAM 7](#_Toc184978337)

[2.6. CÁC CÔNG CỤ HỔ TRỢ 7](#_Toc184978338)

[2.6.1 Website 7](#_Toc184978339)

[2.6.1.1 Mô tả chi tiết: 7](#_Toc184978340)

[2.6.1.2 Điểm mạnh: 8](#_Toc184978341)

[2.6.2 Visual studio code(VS Code) 8](#_Toc184978342)

[2.6.2.1 Mô tả chi tiết: 8](#_Toc184978343)

[2.6.2.2 Điểm mạnh: 8](#_Toc184978344)

[2.6.3 Draw.io 8](#_Toc184978345)

[2.6.3.1 Mô tả chi tiết: 8](#_Toc184978346)

[2.6.3.2 Điểm mạnh: 9](#_Toc184978347)

[2.6.4 Github 9](#_Toc184978348)

[2.6.4.1 Mô tả chi tiết: 9](#_Toc184978349)

[2.6.4.2 Điểm mạnh: 9](#_Toc184978350)

[CHƯƠNG 3: KẾT QUẢ ỨNG DỤNG 10](#_Toc184978351)

[3.1 VẬN DỤNG LÝ THUYẾT VÀO ỨNG DỤNG 10](#_Toc184978352)

[3.1.1 Giải thuật quy hoạch động 10](#_Toc184978353)

[3.1.2 Lưu đồ giải thuật 15](#_Toc184978354)

[3.1.3 Giải thuật tham lam 16](#_Toc184978355)

[3.1.4 Lưu đồ giải thuật tham lam 18](#_Toc184978356)

[3.2 CHƯƠNG TRÌNH DEMO 19](#_Toc184978357)

[3.3 KẾT LUẬN 22](#_Toc184978358)

[3.3.1 So sánh chi tiết hai giải thuật 22](#_Toc184978359)

[3.3.2 Liên hệ thực tế 23](#_Toc184978360)

[3.3.3 Kết luận 24](#_Toc184978361)

[3.3.4 Hướng phát triển 24](#_Toc184978362)

[3.3.5 Ưu điểm, hạn chế - nguyên nhân 25](#_Toc184978363)

[TÀI LIỆU THAM KHẢO 26](#_Toc184978364)

# MỤC LỤC BẢNG

[Bảng 3. 1 Bảng giá trị và trọng lượng từng đồ vật 10](#_Toc184978282)

[Bảng 3. 2 Bảng vòng lặp ở trọng lượng thứ nhất của balo 10](#_Toc184978283)

[Bảng 3. 3 Bảng vòng lặp ở trọng lượng thứ 2 của balo 10](#_Toc184978284)

[Bảng 3. 4 Bảng vòng lặp ở trọng lượng thứ 3 của balo 11](#_Toc184978285)

[Bảng 3. 5 Bảng vòng lặp ở trọng lượng thứ 4 của balo 11](#_Toc184978286)

[Bảng 3. 6 Bảng vòng lặp ở trọng lượng thứ 5 của balo 12](#_Toc184978287)

[Bảng 3. 7 Bảng vòng lặp ở trọng lượng thứ 6 của balo 12](#_Toc184978288)

[Bảng 3. 8 Bảng vòng lặp ở trọng lượng thứ 7 của balo 12](#_Toc184978289)

[Bảng 3. 9 Bảng vòng lặp ở trọng lượng thứ 8 của balo 13](#_Toc184978290)

[Bảng 3. 10 Bảng vòng lặp ở trọng lượng thứ 9 của balo 13](#_Toc184978291)

[Bảng 3. 11 Bảng vòng lặp ở trọng lượng thứ 10 của balo 13](#_Toc184978292)

[Bảng 3. 12 Bảng kết quả cuối cùng khi lặp qua hết trọng lượng để lựa chọn đồ vật 14](#_Toc184978293)

[Bảng 3. 13 Bảng kết quả sau khi chọn được đồ vật 14](#_Toc184978294)

[Bảng 3. 14 Bảng giá trị và trọng lượng từng đồ vật 16](#_Toc184978295)

[Bảng 3. 15 Bảng các đồ vật đã được sắp xếp theo giá trị giảm dần 16](#_Toc184978296)

[Bảng 3. 16 Bảng kết quả cuối cùng khi lặp qua hết trọng lượng để lựa chọn đồ vật 17](#_Toc184978297)

[Bảng 3. 17 Bảng kết quả sau khi chọn được đồ vật 17](#_Toc184978298)

[Bảng 3. 18 Bảng so sánh chi tiết của hai giải thuật 22](#_Toc184978299)

[Bảng 3. 19 Bảng so sánh các trường hợp thực tế của hai giải thuật 24](#_Toc184978300)

# MỤC LỤC HÌNH ẢNH

[Hình 3. 1 Lưu đồ giải thuật quy hoạch động 15](#_Toc184978243)

[Hình 3. 2 Lưu đồ giải thuật tham lam 18](#_Toc184978244)

[Hình 3. 3 From nhập số đồ vật và trọng lượng balo 19](#_Toc184978245)

[Hình 3. 4 Thông báo nếu người dùng nhập số đồ vật và trọng lượng không hợp lệ 20](#_Toc184978246)

[Hình 3. 5 From nhập giá trị - Trọng lượng từng đồ vật 20](#_Toc184978247)

[Hình 3. 6 Thông báo nếu người dùng nhập vào không hợp lệ 20](#_Toc184978248)

[Hình 3. 7 Hình ảnh nếu người dùng nhập thông số hợp lệ 20](#_Toc184978249)

[Hình 3. 8 From giá trị và trọng lượng nếu người dùng nhập đầy đủ 21](#_Toc184978250)

[Hình 3. 9 Màn hình giao diện bảng phương án của hai giải thuật 21](#_Toc184978251)

[Hình 3. 10 Màn hình giao diện kết quả của hai giải thuật 22](#_Toc184978252)

# CHƯƠNG 1: TỔNG QUAN

🖎🕮✍

## 1.1 LỜI MỞ ĐẦU

Bài toán cái balo (Knapsack Problem) là một trong những bài toán tối ưu hóa cổ điển, có ứng dụng rộng rãi trong nhiều lĩnh vực như quản lý tài nguyên, vận tải, kinh doanh và lập kế hoạch. Đây là một bài toán nhằm tìm ra cách tối ưu nhất để chọn các đồ vật từ một tập hợp cho trước sao cho tổng giá trị lớn nhất, đồng thời không vượt quá giới hạn về trọng lượng hoặc kích thước của balo.

Trong bài toán mà em thực hiện, điểm đặc biệt là một đồ vật có thể được chọn nhiều lần. Điều này khiến bài toán trở nên linh hoạt và thực tế hơn, vì trong thực tế, các tài nguyên hoặc sản phẩm có thể không bị giới hạn về số lượng. Ví dụ, nếu có một món đồ với giá trị cao nhưng trọng lượng nhỏ, việc chọn món đồ này nhiều lần sẽ giúp tối đa hóa giá trị tổng thể của balo.

Báo cáo này tập trung vào hai phương pháp giải bài toán:

Phương pháp tham ăn (Greedy): Chọn đồ vật dựa trên giá trị cao nhất tại mỗi bước, ưu tiên những món có lợi thế tối ưu.

Phương pháp quy hoạch động (Dynamic Programming): Xây dựng giải pháp dựa trên các kết quả trung gian và tối ưu hóa bằng cách phân tích toàn bộ không gian trạng thái.

Mục tiêu của báo cáo là tìm hiểu và áp dụng các thuật toán để chọn lựa đồ vật một cách tối ưu nhất, đồng thời so sánh kết quả giữa hai phương pháp trên. Các kết quả sẽ được trình bày thông qua giao diện trực quan, giúp dễ dàng nhận diện được sự khác biệt và hiệu quả của từng cách tiếp cận.

Hy vọng rằng báo cáo này không chỉ giúp làm rõ bài toán cái balo trong trường hợp được phép chọn nhiều lần mà còn cung cấp góc nhìn sâu sắc về các phương pháp tối ưu hóa, từ đó mở rộng khả năng ứng dụng vào các bài toán thực tế khác trong cuộc sống.

## 1.2 MÔ TẢ BÀI TOÁN

Hãy tưởng tượng bạn là một người leo núi, chuẩn bị cho một chuyến thám hiểm kéo dài. Bạn cần mang theo một balo chứa các vật dụng cần thiết, nhưng chiếc balo chỉ có khả năng chịu được một trọng lượng tối đa. Nhiệm vụ của bạn là chọn các vật dụng sao cho tổng giá trị của chúng là cao nhất, đồng thời nếu vượt quá trọng lượng mà balo thì bạn sẻ không thể lên đến đỉnh núi.

Bạn có một chiếc balo với sức chứa tối đa 20kg. Trước mặt bạn là một danh sách các đồ vật cần thiết, mỗi món có trọng lượng và giá trị như sau:

**Đồ vật 1**: Trọng lượng 5kg, Giá trị 8

**Đồ vật 2**: Trọng lượng 7kg, Giá trị 10

**Đồ vật 3**: Trọng lượng 4kg, Giá trị 6

**Đồ vật 4**: Trọng lượng 3kg, Giá trị 5

**Đồ vật 5**: Trọng lượng 2kg, Giá trị 4

**Đồ vật 6**: Trọng lượng 6kg, Giá trị 7

Hãy tối ưu hóa giá trị các đồ vật mang theo sao cho tổng trọng lượng không vượt quá 20kg.

## 1.3 MỤC TIÊU CẦN ĐẠT ĐƯỢC

Mục tiêu của bài toán không chỉ dừng lại ở việc tìm ra giải pháp tối ưu cho bài toán cái balo, mà còn hướng đến việc phát triển một chương trình hoàn chỉnh và thân thiện với người dùng. Chúng tôi mong muốn xây dựng một công cụ có thể xử lý bài toán một cách hiệu quả, chính xác và dễ sử dụng, ngay cả đối với những người không có nhiều kinh nghiệm về công nghệ.

Trong đó, chương trình phải đảm bảo tính chính xác tuyệt đối trong việc tìm ra các phương án tối ưu. Các thuật toán được sử dụng không chỉ đáp ứng yêu cầu về kết quả mà còn phải được tối ưu hóa để xử lý dữ liệu lớn mà không ảnh hưởng đến hiệu suất.

Bên cạnh đó, giao diện cũng là một yếu tố quan trọng. Giao diện phải trực quan, rõ ràng và hỗ trợ người dùng nhập liệu một cách dễ dàng. Mọi thông tin đầu vào cần được kiểm tra và xử lý tự động để giảm thiểu lỗi, giúp người dùng có trải nghiệm sử dụng mượt mà và tiện lợi.

Cuối cùng, chương trình không chỉ là một công cụ giải bài toán, mà còn là một sản phẩm phản ánh sự nỗ lực và sáng tạo. Thông qua quá trình nghiên cứu và phát triển, chúng tôi hy vọng cải thiện kỹ năng lập trình, tư duy giải thuật, đồng thời nâng cao kiến thức về các phương pháp tối ưu hóa và ứng dụng thực tế.

## 1.4 HƯỚNG GIẢI QUYẾT

### 1.4.1 Giải thuật quy hoạch động

(Tìm hiểu trước bài tập dãy con có tổng bằng S để hiểu rỏ hơn về quy hoạch động cho bài toán cái balo)

**Khởi tạo mảng dp:**

+ Tạo mảng dp[0..W] với dp[i] là giá trị tối ưu khi trọng lượng tối đa của balo là i.

**Cập nhật mảng dp:**

+ Với mỗi đồ vật, bạn sẽ duyệt qua tất cả các trọng lượng từ **w[i]** đến **W**.

+ Do có thể chọn một đồ vật nhiều lần, bạn sẽ duyệt từ **w[i]** đến **W.**

+ Cập nhật công thức: Max(dp[j], dp[j – w[i]] + v[i])

+ Điều này có nghĩa là bạn có thể chọn đồ vật **i** bao nhiêu lần cũng được, miễn là tổng trọng lượng không vượt quá **j**.

**Kết quả cuối cùng:**

+ Giá trị tối ưu sẽ nằm trong **dp[W]**, tức là giá trị lớn nhất có thể đạt được khi trọng lượng tối đa của balo là **W**.

### 1.4.2 Giải thuật tham lam

**Sắp xếp đồ vật theo giá trị giảm dần:**

+ Đầu tiên, sắp xếp các đồ vật theo giá trị từ cao đến thấp, ưu tiên chọn đồ vật có giá trị cao nhất mà không xét đến tỷ lệ giá trị/trọng lượng.

**Chọn đồ vật:**

+ Duyệt qua danh sách đồ vật đã sắp xếp, nếu đồ vật có thể vừa với trọng lượng còn lại trong balo, chọn nó và giảm trọng lượng còn lại. Nếu không, bỏ qua đồ vật đó và tiếp tục.

## 1.5 PHẠM VỊ NGHIÊN CỨU

### 1.5.1 Ứng dụng thực tế

Đề tài nghiên cứu tập trung vào việc tìm hiểu các ứng dụng thực tiễn của bài toán cái balo trong nhiều lĩnh vực khác nhau như tối ưu hóa việc sử dụng tài nguyên trong các hệ thống công nghệ thông tin.

Ngoài ra, bài toán cái balo còn được ứng dụng trong việc xây dựng các chiến lược tiếp thị, giúp doanh nghiệp xác định các mặt hàng mang lại giá trị cao nhất trong khi vẫn đảm bảo chi phí vận chuyển và lưu trữ hợp lý. Trong khoa học máy tính, bài toán này có vai trò quan trọng trong việc thiết kế thuật toán cho các hệ thống lưu trữ dữ liệu hoặc tối ưu hóa hiệu suất của các phần mềm.

Các công cụ trực quan hóa cũng được sử dụng để minh họa kết quả giải bài toán, giúp người dùng dễ dàng hiểu và phân tích các quyết định tối ưu dựa trên trọng lượng và giá trị của từng đồ vật. Việc ứng dụng bài toán cái balo không chỉ dừng lại ở các lĩnh vực kỹ thuật mà còn mở rộng đến việc ra quyết định hiệu quả trong nhiều khía cạnh của đời sống và công việc.

### 1.5.2 Môi trường triển khai

Chương trình được xây dựng và triển khai trên nền tảng web, sử dụng các công nghệ như JavaScript, HTML, CSS. Quá trình phát triển và thử nghiệm được thực hiện trên các trình duyệt hiện đại như Google Chrome, Mozilla Firefox, và Microsoft Edge, đảm bảo khả năng tương thích và hiệu suất ổn định.

### 1.5.3 Phạm vi dữ liệu dữ liệu

Đầu vào của bài toán cái balo được giới hạn ở dạng danh sách các đồ vật, trong đó mỗi đồ vật được biểu diễn bởi trọng lượng và giá trị tương ứng. Số lượng đồ vật và dung tích của balo được thiết kế phù hợp với khả năng xử lý của chương trình, đảm bảo hiệu suất tối ưu khi thao tác trên các bộ dữ liệu có kích thước từ nhỏ đến trung bình.

## 1.6 KẾ HOẠCH THỰC HIỆN

**Tuần 1, 2:**

- Phân tích yêu cầu của đề tài: Xác định đầu vào, đầu ra và các chức năng cần thiết của bài toán.

- Tìm kiếm tài liệu liên quan đến thuật toán tham lam và quy hoạch động trong bài toán cái balo.

- Giải bài toán mẫu trên giấy để hiểu rõ phương pháp và kiểm tra tính đúng đắn.

- Vẽ lưu đồ chi tiết cho hai giải thuật tham lam và quy hoạch động để làm cơ sở cho việc triển khai.

**Tuần 3, 4, 5**

Backend:

- Cài đặt các cấu trúc dữ liệu cần thiết bằng JavaScript.

- Viết thuật toán tham lam và quy hoạch động.

- Kiểm tra tính đúng đắn của các thuật toán bằng cách nhập dữ liệu thử nghiệm qua công cụ giả lập (trình duyệt).

Frontend:

- Thiết kế giao diện nhập liệu: số món đồ, trọng lượng balo, giá trị và trọng lượng của từng món đồ.

- Các thành phần cần thiết:

+ Form nhập liệu với các thẻ <input> và nút <button> để gữi dữ liệu.

+ Hiển thị kết quả gồm danh sách đồ vật được chọn và giá trị tối ưu.

+ Vẻ giao diện mẫu demo.

+ Sử dụng HTML, CSS và JavaScript để xây dựng giao diện đã vẻ.

**Tuần 6, 7**

Tích hợp:

- Kết nối giữa giao diện và logic thuật toán bằng cách xử lý dữ liệu đầu vào/đầu ra thông qua JavaScript.

- Tạo chức năng kiểm tra và hiển thị thông báo lỗi khi dữ liệu nhập không hợp lệ (ví dụ: trọng lượng âm hoặc 0, giá trị không phải số, v.v.).

Tối ưu hóa:

- Cải thiện giao diện để thân thiện với người dùng (responsive).

- Kiểm tra khả năng hoạt động của chương trình trên nhiều trình duyệt.

**Tuần 8**

Rà soát:

- Kiểm tra toàn bộ chương trình, sửa lỗi phát sinh nếu có.

- Viết tài liệu hướng dẫn sử dụng cho ứng dụng web.

Báo cáo:

- Hoàn thiện báo cáo với các nội dung: Mô tả bài toán, giải thuật, cách triển khai, các bước thử nghiệm và kết quả thu được.

- Đính kèm các hình ảnh minh họa giao diện và kết quả thực hiện trong báo cáo.

# CHƯƠNG II: CỞ SỞ LÝ THUYẾT

🙞 🙢 🕮 🙠 🙜

## 2.1 HTML

- HTML (viết tắt của Hypertext Markup Language hay ngôn ngữ đánh dấu siêu văn bản) là ngôn ngữ đánh dấu được dùng để tạo các trang siêu văn bản được sử dụng trên mạng internet (hay trang web). Gọi là ngôn ngữ đánh dấu là vì HTML sử dụng các thẻ để định nghĩa (hay đánh dấu) các thành phần khác nhau trên trang web.

**-** HTML là mã được dùng để xây dựng nên cấu trúc và nội dung của trang web. Ví dụ, nội dung có thể được cấu thành bởi một loạt các đoạn văn, một danh sách liệt kê, hoặc sử dụng những hình ảnh và bảng biểu. Như tiêu đề, bài viết này sẽ cho bạn những hiểu biết về HTML và chức năng của nó.

- HTML giúp định dạng văn bản trong trang web nhờ các thẻ. Hơn nữa, các thẻ HTML có thể liên kết từ hoặc một cụm từ với các tài liệu khác trên internet. Đa số các thẻ html có dạng thẻ đóng mở.

## 2.2 CSS

- CSS là ngôn ngữ tạo phong cách cho trang web – Cascading Style Sheet language. Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là HTML. Nó có thể điều khiển định dạng của nhiều trang web cùng lúc để tiết kiệm công sức cho người viết web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc, và font chữ.

- CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì một lý do đơn giản. HTML không được thiết kế để gắn tag để giúp định dạng trang web. Bạn chỉ có thể dùng nó để “đánh dấu” lên site.

- Những tag như <font> được ra mắt trong HTML phiên bản 3.2, nó gây rất nhiều rắc rối cho lập trình viên. Vì website có nhiều font khác nhau, màu nền và phong cách khác nhau. Để viết lại code cho trang web là cả một quá trình dài, cực nhọc. Vì vậy, CSS được tạo bởi W3C là để giải quyết vấn đề này.

- Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời. CSS về lý thuyết không có cũng được, nhưng khi đó website sẽ không chỉ là một trang chứa văn bản mà không có gì khác.

## 2.3 JAVASCRIPT

- JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ (small and lightweight). Khi nằm bên trong một môi trường (host environment), JavaScript có thể kết nối tới các object của môi trường đó và cung cấp các cách quản lý chúng (object).

- JavaScript chứa các thư viện tiêu chuẩn cho các object, ví dụ như: Array, Date, và Math, và các yếu tố cốt lõi của ngôn ngữ lập trình như: toán tử (operators), cấu trúc điều khiển (control structures), và câu lệnh. JavaScript có thể được mở rộng cho nhiều mục đích bằng việc bổ sung thêm các object; ví dụ:

+ Client-side JavaScript - JavaScript phía máy khách, JavaScript được mở rộng bằng cách cung cấp các object để quản lý trình duyệt và Document Object Model (DOM) của nó. Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.

+ Server-side JavaScript - JavaScript phía Server, JavaScript được mở rộng bằng cách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.

## 2.4 THUẬT TOÁN QUY HOẠCH ĐỘNG

Quy hoạch động (**Dynamic Programming**, viết tắt là **DP**) là một phương pháp giải quyết các bài toán tối ưu có cấu trúc con chồng chéo (overlapping subproblems) và tính chất con tối ưu (optimal substructure). Thay vì giải quyết lại các vấn đề con giống nhau nhiều lần, quy hoạch động sẽ lưu trữ kết quả của các vấn đề con đã tính toán và sử dụng lại chúng khi cần, giúp tối ưu hóa thời gian tính toán.

Ưu điểm:

**- Giải quyết bài toán tối ưu**: DP chia bài toán lớn thành các bài toán con nhỏ hơn, giải quyết từng bài toán con và lưu trữ kết quả để tránh tính lại nhiều lần, từ đó tối ưu hóa quá trình tính toán.

**- Giảm độ phức tạp thời gian**: Lưu trữ kết quả của các bài toán con đã giải quyết giúp giảm số lần tính toán, giảm đáng kể độ phức tạp so với phương pháp brute force.

**- Tìm giải pháp tối ưu**: DP luôn cho ra giải pháp tối ưu cho các bài toán tối ưu (ví dụ: bài toán cái balo, chuỗi con dài nhất chung, v.v...) nếu bài toán có tính chất tối ưu con và chồng lặp con.

**- Áp dụng rộng rãi**: DP có thể áp dụng cho nhiều bài toán khác nhau như chuỗi con, phân hoạch, tìm kiếm tối ưu trong đồ thị, v.v...

**- Đảm bảo tính chính xác**: DP tính toán từng bước một cách cẩn thận, đảm bảo tính chính xác cho kết quả cuối cùng.

Nhược điểm:

**- Tiêu tốn bộ nhớ**: Quy hoạch động thường yêu cầu lưu trữ các kết quả trung gian của các bài toán con, điều này có thể tiêu tốn một lượng bộ nhớ lớn, đặc biệt khi bài toán có kích thước lớn.

**- Cài đặt phức tạp**: So với các thuật toán đơn giản khác như tham lam hay chia để trị, DP yêu cầu việc xác định các trạng thái và các mối quan hệ giữa chúng, khiến cho việc cài đặt trở nên phức tạp hơn.

**- Không thích hợp cho mọi bài toán**: DP chỉ hiệu quả với những bài toán có tính chất **tối ưu con** và **chồng lặp con**. Nếu bài toán không có những tính chất này, việc sử dụng DP sẽ không mang lại hiệu quả và có thể gây lãng phí tài nguyên.

**- Khó tối ưu bộ nhớ**: Trong một số trường hợp, mặc dù bài toán có thể giải quyết bằng DP, việc tối ưu hóa bộ nhớ để lưu trữ kết quả lại là một vấn đề phức tạp. Bạn phải tìm cách giảm bớt bộ nhớ mà vẫn giữ được kết quả cần thiết, điều này có thể làm tăng độ phức tạp trong cài đặt.

**- Khó khăn trong việc kiểm tra tính đúng đắn**: Kiểm tra tính đúng đắn của một giải pháp DP có thể gặp khó khăn vì bạn không chỉ cần kiểm tra kết quả cuối cùng mà còn phải kiểm tra tất cả các bước tính toán trung gian.

## 2.5 THUẬT TOÁN THAM LAM

Bài toán tham lam (greedy algorithm) là một phương pháp giải quyết bài toán trong đó ta đưa ra quyết định từng bước một, mỗi bước chọn lựa tối ưu tại thời điểm đó, với hy vọng rằng các quyết định cục bộ tối ưu sẽ dẫn đến một giải pháp tối ưu toàn cục.Ưu điểm: Nhanh, dễ triển khai.

Ưu điểm:

**- Đơn giản và dễ hiểu**: Thuật toán tham lam có cấu trúc đơn giản, dễ cài đặt, với mỗi bước chọn lựa tối ưu tại thời điểm mà không cần xem lại các quyết định trước.

**- Hiệu quả về thời gian**: Thuật toán tham lam thường tìm ra giải pháp nhanh chóng mà không cần kiểm tra tất cả các khả năng, giảm độ phức tạp thời gian.

**- Tiết kiệm bộ nhớ**: Không yêu cầu lưu trữ nhiều thông tin như quy hoạch động, giúp tiết kiệm bộ nhớ.

**- Dễ triển khai:** Dễ dàng cài đặt mà không cần các bảng hay mảng phức tạp.

Nhược điểm:

**- Không luôn cho giải pháp tối ưu**: Tham lam không đảm bảo giải pháp tối ưu, chỉ đảm bảo lựa chọn tốt nhất tại mỗi bước, nhưng không chắc chắn rằng các lựa chọn này sẽ dẫn đến giải pháp tối ưu cuối cùng.

**- Phụ thuộc vào bài toán**: Tham lam chỉ hiệu quả với bài toán có tính chất "tối ưu con". Nếu bài toán không có tính chất này, thuật toán có thể không cho ra giải pháp đúng.

**- Không xem xét toàn bộ bài toán**: Tham lam chỉ tập trung vào lựa chọn tối ưu trong từng bước mà không xét đến các lựa chọn tương lai, có thể bỏ qua các khả năng tốt hơn.

**- Thiếu khả năng "quay lại"**: Nếu chọn nhầm bước, thuật toán tham lam không thể điều chỉnh lại, như trong các thuật toán quy hoạch động hay chia để trị.

## 2.6. CÁC CÔNG CỤ HỔ TRỢ

### 2.6.1 Website

#### 2.6.1.1 Mô tả chi tiết:

Website tĩnh là một loại trang web có nội dung cố định, không yêu cầu xử lý từ phía máy chủ. Các trang web tĩnh thường được xây dựng bằng HTML, CSS và JavaScript cơ bản, và khi người dùng truy cập, trình duyệt sẽ hiển thị nội dung trực tiếp từ tệp được lưu trữ. Đây là giải pháp lý tưởng cho các trang trình bày thông tin đơn giản hoặc giao diện mẫu.

#### 2.6.1.2 Điểm mạnh:

**- Dễ triển khai và bảo trì:** Không cần cấu hình phức tạp hay cơ sở dữ liệu, phù hợp cho các dự án nhỏ hoặc cá nhân.

**- Tốc độ tải nhanh:** Do không phải xử lý phía máy chủ, nội dung được tải trực tiếp từ tệp, giảm thời gian chờ đợi.

**- Chi phí thấp:** Có thể lưu trữ miễn phí trên các nền tảng như GitHub Pages, Netlify hoặc Vercel.

**- Bảo mật cao:** Không có cơ sở dữ liệu nên tránh được nhiều rủi ro bảo mật như SQL Injection.

### 2.6.2 Visual studio code(VS Code)

#### 2.6.2.1 Mô tả chi tiết:

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn nhẹ nhưng mạnh mẽ, được phát triển bởi Microsoft. Với giao diện thân thiện, hỗ trợ đa nền tảng (Windows, macOS, Linux), đây là công cụ phổ biến nhất cho lập trình viên hiện nay. VS Code hỗ trợ rất nhiều ngôn ngữ lập trình như JavaScript, Python, PHP, C++, và tích hợp mạnh mẽ với Git.

#### 2.6.2.2 Điểm mạnh:

**- Giao diện thân thiện:** Giao diện tối giản nhưng dễ sử dụng, giúp lập trình viên tập trung vào công việc chính.

**- Tùy biến cao:** Hỗ trợ thay đổi giao diện, phím tắt và tích hợp các tiện ích mở rộng theo nhu cầu cá nhân.

**- Hệ sinh thái Extensions phong phú:** Tích hợp hàng ngàn tiện ích mở rộng miễn phí từ kiểm tra lỗi, tự động hoàn thành mã (IntelliSense), hỗ trợ ngôn ngữ lập trình mới, đến tích hợp framework.

**- Hỗ trợ gỡ lỗi (Debugging):** VS Code có công cụ gỡ lỗi trực tiếp cho nhiều ngôn ngữ lập trình, giúp phát hiện lỗi nhanh chóng.

**- Tích hợp Git:** Dễ dàng quản lý phiên bản ngay trong giao diện VS Code mà không cần chuyển đổi sang công cụ khác.

**- Hiệu suất vượt trội:** Tuy là công cụ mạnh mẽ nhưng VS Code rất nhẹ, phù hợp cho cả máy cấu hình yếu.

### 2.6.3 Draw.io

#### 2.6.3.1 Mô tả chi tiết:

Draw.io là công cụ trực tuyến mạnh mẽ, miễn phí, dùng để vẽ sơ đồ và lưu đồ. Đây là một phần mềm phổ biến cho các nhà thiết kế hệ thống, lập trình viên hoặc bất kỳ ai cần trực quan hóa ý tưởng, mô hình hoặc quy trình làm việc. Công cụ này hỗ trợ làm việc trên trình duyệt, có thể cài đặt độc lập hoặc tích hợp với các dịch vụ lưu trữ đám mây như ogle Drive, OneDrive.

#### 2.6.3.2 Điểm mạnh:

**- Miễn phí:** Không yêu cầu chi phí hoặc tài khoản để sử dụng, phù hợp cho mọi đối tượng.

**- Giao diện kéo-thả dễ sử dụng:** Không yêu cầu kỹ năng phức tạp, bất kỳ ai cũng có thể dễ dàng tạo sơ đồ.

**- Hỗ trợ đa dạng mẫu:** Bao gồm các loại sơ đồ UML, ERD, biểu đồ luồng, sơ đồ tổ chức và nhiều hơn nữa.

**- Tích hợp mạnh mẽ:** Cho phép lưu và chỉnh sửa trực tiếp trên các nền tảng đám mây, chia sẻ dễ dàng với nhóm làm việc.

**- Xuất file linh hoạt:** Hỗ trợ các định dạng phổ biến như PNG, PDF, SVG hoặc định dạng gốc XML để dễ dàng chỉnh sửa sau này.

### 2.6.4 Github

#### 2.6.4.1 Mô tả chi tiết:

GitHub là nền tảng quản lý mã nguồn và hợp tác làm việc dựa trên Git, giúp các nhà phát triển lưu trữ, quản lý phiên bản mã nguồn và làm việc nhóm hiệu quả. Ngoài ra, GitHub còn là nơi chia sẻ mã nguồn mở lớn nhất thế giới, cung cấp môi trường để học hỏi và phát triển dự án từ cộng đồng lập trình viên.

#### 2.6.4.2 Điểm mạnh:

**- Quản lý phiên bản hiệu quả:** Giúp theo dõi lịch sử thay đổi mã nguồn, dễ dàng hoàn tác các thay đổi không mong muốn.

**- Hỗ trợ làm việc nhóm:** Cho phép nhiều lập trình viên cộng tác trên cùng một dự án, sử dụng hệ thống phân nhánh (branch) và hợp nhất (merge).

**- Tích hợp CI/CD:** Hỗ trợ triển khai tự động, kiểm tra và xây dựng dự án trực tiếp trên GitHub.

**- Bảo mật cao:** Cung cấp tùy chọn kho lưu trữ riêng tư cho dự án quan trọng.

**- Hỗ trợ trình bày dự án:** Có thể tạo website trình bày thông tin dự án thông qua GitHub Pages.

**- Nền tảng học tập:** Cộng đồng GitHub là nơi các lập trình viên chia sẻ mã nguồn và học hỏi kinh nghiệm từ những dự án lớn.

# CHƯƠNG 3: KẾT QUẢ ỨNG DỤNG

🙞 🙢 🕮 🙠 🙜

## 3.1 VẬN DỤNG LÝ THUYẾT VÀO ỨNG DỤNG

### 3.1.1 Giải thuật quy hoạch động

Có 4 đồ vật, trọng lượng tối đa 10kg

|  |  |
| --- | --- |
| W | V |
| 1 | 1 |
| 3 | 4 |
| 4 | 5 |
| 6 | 8 |

Bảng 3. 1 Bảng giá trị và trọng lượng từng đồ vật

**Bước 1:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |  |  |  |  |  |  |  |  |  |
| 3 | 4 | 2 | 0 | 1 |  |  |  |  |  |  |  |  |  |
| 4 | 5 | 3 | 0 | 1 |  |  |  |  |  |  |  |  |  |
| 6 | 8 | 4 | 0 | 1 |  |  |  |  |  |  |  |  |  |

Bảng 3. 2 Bảng vòng lặp ở trọng lượng thứ nhất của balo

Đồ vật 1

Max(GTLN[j], GTLN[j – w[i]] + v[i])

GTLN[1]: GTLN ở cột 1 là 0

GTLN[1-1] = GTLN[0] ta xem ở cột 0 và GTLN của nó là 0, ta lấy:

0 + v[i] = 0 + 1 = 1

Max(0, 1) => 1

Các đồ vật còn lại đề k thỏa điều kiện >= j => viết lại GTLN

**Bước 2:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 2 |  |  |  |  |  |  |  |  |
| 3 | 4 | 2 | 0 | 1 | 2 |  |  |  |  |  |  |  |  |
| 4 | 5 | 3 | 0 | 1 | 2 |  |  |  |  |  |  |  |  |
| 6 | 8 | 4 | 0 | 1 | 2 |  |  |  |  |  |  |  |  |

Bảng 3. 3 Bảng vòng lặp ở trọng lượng thứ 2 của balo

Đồ vật 1

Max(GTLN[j], GTLN[j – w[i]] + v[i])

GTLN[2]: GTLN ở cột 2 là 0

GTLN[2-1] = GTLN[1] ta xem ở cột 1 và GTLN của nó là 1, ta lấy:

1 + v[i] = 1 + 1 = 2

Max(0, 2) => 2

Các đồ vật còn lại đề k thỏa điều kiện >= j => viết lại GTLN

**Bước 3:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 2 | 3 |  |  |  |  |  |  |  |
| 3 | 4 | 2 | 0 | 1 | 2 | 4 |  |  |  |  |  |  |  |
| 4 | 5 | 3 | 0 | 1 | 2 | 4 |  |  |  |  |  |  |  |
| 6 | 8 | 4 | 0 | 1 | 2 | 4 |  |  |  |  |  |  |  |

Bảng 3. 4 Bảng vòng lặp ở trọng lượng thứ 3 của balo

Đồ vật 1

Max(GTLN[j], GTLN[j – w[i]] + v[i])

GTLN[3]: GTLN ở cột 3 là 0

GTLN[3 - 1] = GTLN[2] ta xem ở cột 2 và GTLN của nó là 2, ta lấy:

0 + v[i] = 2 + 1 = 3

Max(0, 3) => 3

Đồ vật 2

Max(GTLN[j], GTLN[j – w[i]] + v[i])

GTLN[3]: GTLN ở cột 3 là 3

GTLN[3 - 3] = GTLN[0] ta xem ở cột 0 và GTLN của nó là 0, ta lấy:

0 + v[i] = 0 + 4 = 4

Max(3, 4) => 4

Các đồ vật còn lại đề k thỏa điều kiện >= j => viết lại GTLN

**Bước 4:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 2 | 3 | 5 |  |  |  |  |  |  |
| 3 | 4 | 2 | 0 | 1 | 2 | 4 | 5 |  |  |  |  |  |  |
| 4 | 5 | 3 | 0 | 1 | 2 | 4 | 5 |  |  |  |  |  |  |
| 6 | 8 | 4 | 0 | 1 | 2 | 4 | 5 |  |  |  |  |  |  |

Bảng 3. 5 Bảng vòng lặp ở trọng lượng thứ 4 của balo

Đồ vật 1

Max(GTLN[4], GTLN[4-1] + 1) => Max(0, 4 + 1) => 5

Đồ vật 2

Max(GTLN[4], GTLN[4-3] + 4) => Max(5, 1 + 4) => 5 (Giữ nguyên chỉ lấy lớn hơn)

Đồ vật 3

Max(GTLN[4], GTLN[4-4] + 5) => Max(5, 0 + 5) => 5

**Bước 5:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 2 | 3 | 5 | 6 |  |  |  |  |  |
| 3 | 4 | 2 | 0 | 1 | 2 | 4 | 5 | 6 |  |  |  |  |  |
| 4 | 5 | 3 | 0 | 1 | 2 | 4 | 5 | 6 |  |  |  |  |  |
| 6 | 8 | 4 | 0 | 1 | 2 | 4 | 5 | 6 |  |  |  |  |  |

Bảng 3. 6 Bảng vòng lặp ở trọng lượng thứ 5 của balo

Đồ vật 1

Max(GTLN[5], GTLN[5-1] + 1) => Max(0, 5 + 1) => 6

Đồ vật 2

Max(GTLN[5], GTLN[5-3] + 4) => Max(6, 2 + 4) => 6

Đồ vật 3

Max(GTLN[5], GTLN[5-4] + 1) => Max(6, 1 + 5) => 6

**Bước 6:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 2 | 3 | 5 | 6 | 7 |  |  |  |  |
| 3 | 4 | 2 | 0 | 1 | 2 | 4 | 5 | 6 | 8 |  |  |  |  |
| 4 | 5 | 3 | 0 | 1 | 2 | 4 | 5 | 6 | 8 |  |  |  |  |
| 6 | 8 | 4 | 0 | 1 | 2 | 4 | 5 | 6 | 8 |  |  |  |  |

Bảng 3. 7 Bảng vòng lặp ở trọng lượng thứ 6 của balo

Đồ vật 1

Max(GTLN[6], GTLN[6-1] + 1) => Max(0, 6 + 1) => 7

Đồ vật 2

Max(GTLN[6], GTLN[6-3] + 4) => Max(7, 4 + 4) => 8

Đồ vật 3

Max(GTLN[6], GTLN[6-4] + 5) => Max(8, 2 + 5) => 8

Đồ vật 4

Max(GTLN[6], GTLN[6-6] + 8) => Max(8, 0 + 8) => 8

**Bước 7:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 2 | 3 | 5 | 6 | 7 | 9 |  |  |  |
| 3 | 4 | 2 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 |  |  |  |
| 4 | 5 | 3 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 |  |  |  |
| 6 | 8 | 4 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 |  |  |  |

Bảng 3. 8 Bảng vòng lặp ở trọng lượng thứ 7 của balo

Đồ vật 1

Max(GTLN[7], GTLN[7-1] + 1) => Max(0, 8 + 1) => 9

Đồ vật 2

Max(GTLN[7], GTLN[7-3] + 4) => Max(9, 5+ 4) => 9

Đồ vật 3

Max(GTLN[7], GTLN[7-4] + 5) => Max(9, 4+ 5) => 9

Đồ vật 4

Max(GTLN[7], GTLN[7-6] + 8) => Max(9, 1+ 8) => 9

**Bước 8:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 2 | 3 | 5 | 6 | 7 | 9 | 10 |  |  |
| 3 | 4 | 2 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 |  |  |
| 4 | 5 | 3 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 |  |  |
| 6 | 8 | 4 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 |  |  |

Bảng 3. 9 Bảng vòng lặp ở trọng lượng thứ 8 của balo

Đồ vật 1

Max(GTLN[8], GTLN[8-1] + 1) => Max(0, 9 + 1) => 10

Đồ vật 2

Max(GTLN[8], GTLN[8-3] + 4) => Max(10, 6+ 4) => 10

Đồ vật 3

Max(GTLN[8], GTLN[8-4] + 5) => Max(10, 5+ 5) => 10

Đồ vật 4

Max(GTLN[8], GTLN[8-6] + 8) => Max(10, 2+ 8) => 10

**Bước 9:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 2 | 3 | 5 | 6 | 7 | 9 | 10 | 11 |  |
| 3 | 4 | 2 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 12 |  |
| 4 | 5 | 3 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 12 |  |
| 6 | 8 | 4 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 12 |  |

Bảng 3. 10 Bảng vòng lặp ở trọng lượng thứ 9 của balo

Đồ vật 1

Max(GTLN[9], GTLN[9-1] + 1) => Max(0, 10 + 1) => 11

Đồ vật 2

Max(GTLN[9], GTLN[9-3] + 4) => Max(10, 8 + 4) => 12

Đồ vật 3

Max(GTLN[9], GTLN[9-4] + 5) => Max(10, 6 + 5) => 12

Đồ vật 4

Max(GTLN[9], GTLN[9-6] + 8) => Max(10, 4 + 8) => 12

**Bước 10:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 2 | 3 | 5 | 6 | 7 | 9 | 10 | 11 | 13 |
| 3 | 4 | 2 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 12 | 13 |
| 4 | 5 | 3 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 12 | 13 |
| 6 | 8 | 4 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 12 | 13 |

Bảng 3. 11 Bảng vòng lặp ở trọng lượng thứ 10 của balo

Đồ vật 1

Max(GTLN[10], GTLN[10-1] + 1) => Max(0, 12 + 1) => 13

Đồ vật 2

Max(GTLN[10], GTLN[10-3] + 4) => Max(10, 9 + 4) => 13

Đồ vật 3

Max(GTLN[10], GTLN[10-4] + 5) => Max(10, 8 + 5) => 13

Đồ vật 4

Max(GTLN[10], GTLN[10-6] + 8) => Max(10, 5 + 8) => 13

**Bước 11:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| $ | W | V | i\j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1.00 | 1 | 1 | 1 | 0 | 1 | 2 | 3 | 5 | 6 | 7 | 9 | 10 | 11 | 13 |
| 1.33 | 3 | 4 | 2 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 12 | 13 |
| 1.25 | 4 | 5 | 3 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 12 | 13 |
| 1.33 | 6 | 8 | 4 | 0 | 1 | 2 | 4 | 5 | 6 | 8 | 9 | 10 | 12 | 13 |

Bảng 3. 12 Bảng kết quả cuối cùng khi lặp qua hết trọng lượng để lựa chọn đồ vật

Ta được:

- GTLN là 13 đang ở đồ vật 1 (cùng hàng) => Chọn đồ vật 1

- Ta lấy 10 – 1 (v[1]) = 9, Xem ở cột 8 => Chọn đồ vật 2

- Ta lấy 9 – 3 (v[1]) = 6, Xem ở cột 6 => Chọn đồ vật 2

- Ta lấy 6 – 3 (v[1]) = 3, Xem ở cột 3 => Chọn đồ vật 2

- Ta lấy 3 – 3 = 0. Kết thúc

**Kết luận:**

|  |  |
| --- | --- |
| **Tổng giá trị** | 13 |
| **Đồ vật được chọn** | Đồ vật 1 chọn 1 lần  Đồ vật 2 chọn 3 lần |
| **Trọng lượng còn lại** | 0 kg |
| **Tổng đơn giá** | 5$ |

Bảng 3. 13 Bảng kết quả sau khi chọn được đồ vật

### 3.1.2 Lưu đồ giải thuật

A diagram of a flowchart

Description automatically generated

Hình 3. 1 Lưu đồ giải thuật quy hoạch động

Mô tả lưu đồ giải thuât:

- Khởi tạo thuật toán: Bắt đầu bằng việc khởi tạo thuật toán với giá trị ban đầu của biến chỉ số i = 0, đại diện cho việc duyệt qua từng món đồ trong danh sách.

- Kiểm tra điều kiện duyệt món đồ **i <= số món đồ**:

+ Nếu **đúng**, tiếp tục xử lý món đồ hiện tại.

+ Nếu **sai**, nghĩa là đã duyệt qua tất cả các món đồ, thuật toán kết thúc.

- Duyệt qua các trọng lượng balo: Khởi tạo chỉ số j = 0, đại diện cho từng mức trọng lượng của balo từ nhỏ nhất đến tối đa.

- Kiểm tra điều kiện trọng lượng balo: **j <= trọng lượng balo**:

+ Nếu **đúng**, tiếp tục xử lý giá trị tại trọng lượng j.

+ Nếu **sai**, tăng i để duyệt món đồ tiếp theo.

- Kiểm tra khả năng đưa món đồ vào balo: Tại mỗi mức trọng lượng, thuật toán kiểm tra xem **trọng lượng món đồ W[i] <= trọng lượng j**:

+ Nếu **đúng**, xem xét giá trị tối ưu khi đưa món đồ này vào balo.

+ Nếu **sai**, bỏ qua món đồ này và tăng j để xét trọng lượng tiếp theo.

- Cập nhật giá trị tối ưu: Khi món đồ có thể được đưa vào balo, thuật toán so sánh giá trị tối ưu tại trọng lượng j hiện tại với giá trị mới nếu thêm món đồ vào. Điều kiện kiểm tra là:

+ Nếu **giá trị hiện tại < giá trị khi thêm món đồ**, cập nhật giá trị tối ưu bằng cách thêm giá trị của món đồ và giảm trọng lượng tương ứng. Đồng thời, lưu lại việc chọn món đồ này.

- Lặp lại quá trình: Sau khi xử lý xong một trọng lượng, thuật toán lặp lại để xét các trọng lượng và món đồ còn lại, cho đến khi tất cả các trường hợp đều được xét.

- Kết thúc thuật toán: Khi đã duyệt hết các món đồ và trọng lượng, thuật toán hoàn thành. Giá trị tối ưu của balo được trả về là giá trị lớn nhất thu được.

### 3.1.3 Giải thuật tham lam

Có 4 đồ vật, trọng lượng tối đa 10kg

|  |  |
| --- | --- |
| W | V |
| 1 | 1 |
| 3 | 4 |
| 4 | 5 |
| 6 | 8 |

Bảng 3. 14 Bảng giá trị và trọng lượng từng đồ vật

**Bước 1**: Sắp xếp các đồ vật theo giá trị (V) giảm dần

|  |  |  |
| --- | --- | --- |
| W | V | Đồ vật |
| 6 | 8 | 4 |
| 4 | 5 | 3 |
| 3 | 4 | 2 |
| 1 | 1 | 1 |

Bảng 3. 15 Bảng các đồ vật đã được sắp xếp theo giá trị giảm dần

**Bước 2:** Chọn đồ vật bỏ vào balo

Đồ vật 4

While(Tổng trọng lượng + Trọng lượng đồ vật 4 <= Trọng lượng Balo)

0 + 6 = 6 <= 10 => Chọn đồ vật 4

Tổng trọng lượng += Trọng lượng đồ vật 4 = 0 + 6 = 6

6 + 6 = 12 <= 10 => Dừng

Đồ vật 3

While(Tổng trọng lượng + Trọng lượng đồ vật 3 <= Trọng lượng Balo)

6 + 4 = 10 <= 10 => Chọn đồ vật 3

Tổng trọng lượng += Trọng lượng đồ vật 3 = 6 + 4 = 10

10 + 4 = 14 <= 10 => Dừng

Đồ vật 2

While(Tổng trọng lượng + Trọng lượng đồ vật 2 <= Trọng lượng Balo)

10 + 3 = 13 <= 10 => Dừng

Đồ vật 1:

While(Tổng trọng lượng + Trọng lượng đồ vật 1 <= Trọng lượng Balo)

10 + 1 = 11 <= 10 => Dừng

Ta có bảng chọn đồ vật (không sắp xếp)

|  |  |  |  |
| --- | --- | --- | --- |
| Giá trị | Trọng lượng | Đồ vật | Chọn (lần) |
| 1 | 1 | 1 | 0 |
| 3 | 4 | 2 | 0 |
| 4 | 5 | 3 | 1 |
| 6 | 8 | 4 | 1 |

Bảng 3. 16 Bảng kết quả cuối cùng khi lặp qua hết trọng lượng để lựa chọn đồ vật

**Bước 3** Kết luận:

|  |  |
| --- | --- |
| **Tổng giá trị** | 13 |
| **Đồ vật được chọn** | Đồ vật 4 chọn 1 lần, Đồ vật 3 chọn 1 lần |
| **Trọng lượng còn lại** | 0 kg |
| **Tổng đơn giá** | 2.58$ |

Bảng 3. 17 Bảng kết quả sau khi chọn được đồ vật

### 3.1.4 Lưu đồ giải thuật tham lam

A diagram of a flowchart

Description automatically generated

Hình 3. 2 Lưu đồ giải thuật tham lam

Mô tả lưu đồ giải thuật:

- Khởi tạo thuật toán: Bắt đầu bằng việc khởi tạo các biến:

**+ Tổng giá trị** = 0 (lưu giá trị tối ưu hiện tại).

**+ Tổng trọng lượng** = 0 (tổng trọng lượng các món đồ đã chọn).

**+ Tổng đơn giá** = 0 (nếu cần tính hiệu quả chọn đồ theo giá trị/trọng lượng).

**+ Lần chọn các đồ vật** = 0 (số lần các món đồ được chọn).

+ Biến đếm **i = 0** (để duyệt qua danh sách đồ vật).

- Kiểm tra điều kiện duyệt danh sách đồ vật: **i < độ dài Arr**:

+ Nếu **đúng**, tiếp tục xét món đồ thứ i.

+ Nếu **sai**, nghĩa là đã duyệt hết các món đồ, thuật toán kết thúc.

- Kiểm tra khả năng thêm món đồ vào balo:

**Tổng trọng lượng + trọng lượng đồ vật i < Trọng lượng balo**:

+ Nếu **đúng**, món đồ có thể thêm vào balo.

+ Nếu **sai**, bỏ qua món đồ này và chuyển sang món tiếp theo.

- Cập nhật khi thêm món đồ vào balo: Nếu món đồ thỏa mãn điều kiện:

**+ Tổng giá trị** được cộng thêm giá trị của món đồ i.

**+ Tổng trọng lượng** được cộng thêm trọng lượng của món đồ i.

+ Tăng **số lần chọn đồ vật** (nếu cần đếm số món đồ được chọn).

**+ Lần chọn đồ vật i** được tăng lên (để lưu lại món đồ nào đã chọn).

- Lặp lại quá trình: Tăng giá trị **i++** để duyệt qua món đồ tiếp theo, quay lại bước kiểm tra điều kiện cho đến khi tất cả các món đồ đã được xử lý.

**-** Khi đã duyệt hết danh sách món đồ, thuật toán kết thúc. Kết quả trả về bao gồm:

+ Tổng giá trị tối ưu đã đạt được.

+ Tổng trọng lượng các món đồ trong balo.

+ Danh sách các món đồ đã chọn.

## 3.2 CHƯƠNG TRÌNH DEMO

- Chương trình gồm các chức năng sau:

+ Nhập số lượng đồ vật và trọng lượng tối đa của balo.

+ Nhập lần lượt giá trị và trọng lượng của từng đồ vật.

+ Chương trình sẻ chạy song song hai giải thuật quy hoạch động và tham lam.

+ Hiển thị ra bảng phương án của hai giải thuật.

+ Hiển thị kết quả của hai giải thuật ra ngoài giao diện để so sách kết quả của hai giải thuật.

- Chọn thuật toán cần thực hiện:

**Bước 1**: Nhập số lượng đồ vật và trọng lượng tối đa của balo vào from

A screenshot of a computer

Description automatically generated

Hình 3. 3 From nhập số đồ vật và trọng lượng balo

- Trường hợp nếu người dùng nhập vào số món đồ và trọng lượng balo là 0 hoặc rổng, các giá trị là số âm và số thập phân sẻ có thông báo lỗi.

A white background with red text

Description automatically generated

Hình 3. 4 Thông báo nếu người dùng nhập số đồ vật và trọng lượng không hợp lệ

A yellow and blue circular logo

Description automatically generated- Sau khi nhập đầy đủ các thông tin sẻ nhấn vào nút hoàn thành để tiến hành bước sau.

+ Cùng với đó là nút load lại chương trình sẻ xuất hiện.

+ Bạn muốn tham khảo suorce có thể click vào đây .

**Bước 2**: Nhập lần lượt giá trị và trọng lượng của từng đồ vật (giả sử ta nhập vào 4 đồ vật và trọng lượng tối đa 10kg).

A screenshot of a video game

Description automatically generated

Hình 3. 5 From nhập giá trị - Trọng lượng từng đồ vật

- Tượng tự như bước 1 nều giá trị không hợp lệ sẻ thông báo lỗi và có viền màu đỏ.

A screenshot of a computer

Description automatically generated

Hình 3. 6 Thông báo nếu người dùng nhập vào không hợp lệ

- Nếu bạn nhập vào hợp lệ sẻ có viền màu xanh.

A white rectangular object with a blue border

Description automatically generated

Hình 3. 7 Hình ảnh nếu người dùng nhập thông số hợp lệ

- Giả sữ ta nhập vào giá trị và trọng lượng như sau và nhấn hoàn thành:

A screenshot of a game

Description automatically generated

Hình 3. 8 From giá trị và trọng lượng nếu người dùng nhập đầy đủ

**A screenshot of a calendar

Description automatically generatedBước 3**: Thực hiện tính toán và in ra bảng phương án của hai giải thuật

Hình 3. 9 Màn hình giao diện bảng phương án của hai giải thuật

+ Đồ vật và i: là số thứ tự của các đồ vật.

+ : là giá trị của đồ vật thứ (i).

+ j : là mỏi kg của trọng lượng balo.

+ : là đơn giá của đồ vật được tính bằng công thức:

+ : là trọng lượng của đồ vật thứ (i).

+ Các giá trị có màu đỏ bên bảng quy hoạch động là giá lớn nhất ở mỏi trọng lượng mà trương trình đã tình toán các giá trị bảng giá trị trước đó không được xem là giá trị lớn nhất.

+ Các giá trị có màu đỏ bên bảng tham lam là tương ứng với đồ vật đó được chọn (>0).

**Bước 4**: in ra kết quả của hai giải thuật

A cartoon character with a light bulb above his head

Description automatically generated

Hình 3. 10 Màn hình giao diện kết quả của hai giải thuật

- Bên phải là đáp án của giải thuật quy hoạch động bên trái là đáp án của giải thuật tham lam và từ màn hình này ta có thể biết được giải thuật nòa trong trường hợp nào là tối ưu nhất.

## 3.3 KẾT LUẬN

### 3.3.1 So sánh chi tiết hai giải thuật

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Quy hoạch động** | **Tham lam** |
| Ý tưởng | Xét tất cả các tổ hợp đồ vật, chọn phương án tối ưu nhất. | Chọn đồ vật theo tiêu chí cục bộ (tỷ lệ giá trị/trọng lượng tốt nhất). |
| Kết quả | Luôn chính xác và tối ưu toàn cục. | Nhanh nhưng không luôn tối ưu. |
| Độ phức tạp thời gian | O(n×W), với nnn là số đồ vật, WWW là trọng lượng balo. | O(nlogn) (sắp xếp danh sách). |
| Ứng dụng thực tế | Dùng khi cần độ chính xác tuyệt đối. | Nhanh và hiệu quả cho các bài toán đơn giản. |
| Độ chính xác | Đảm bảo giải pháp chính xác 100%. | Có thể không tối ưu trong một số trường hợp. |
| Khả năng mở rộng | Khó mở rộng cho các bài toán lớn với số lượng đồ vật hoặc trọng lượng lớn do hạn chế thời gian và bộ nhớ. | Dễ mở rộng cho các bài toán lớn, miễn là tiêu chí cục bộ hợp lý. |
| Tính dễ triển khai | Phức tạp hơn, cần hiểu rõ thuật toán và cách lưu trữ trạng thái. | Đơn giản hơn, dễ viết và triển khai. |
| Phụ thuộc đầu vào | Không phụ thuộc vào cách sắp xếp ban đầu của danh sách đồ vật. | Phụ thuộc vào thứ tự sắp xếp theo tiêu chí cục bộ (thường là giá trị/trọng lượng). |

Bảng 3. 18 Bảng so sánh chi tiết của hai giải thuật

- Khi nào nên sữ dụng giải thuật tham lam:

+ Số lượng đồ vật lớn và cần tính toán nhanh.

+ Các đồ vật có tỷ lệ giá trị phân bố đồng đều.

+ Thực tế:

* **Đóng gói hàng hóa**: Một công ty vận chuyển cần xếp hàng hóa lên xe tải với khối lượng tối đa. Nếu các kiện hàng có giá trị tương đương nhau, giải thuật Tham lam sẽ nhanh và đủ tốt.
* **Quản lý ngân sách**: Khi phân bổ ngân sách cho nhiều dự án với tỷ lệ lợi nhuận đầu tư giống nhau.

- Khi nào nên sữ dụng giải thuật quy hoạch động:

+ Đồ vật có trọng lượng và giá trị không đồng đều, hoặc tổ hợp tối ưu không rõ ràng.

+ Bài toán yêu cầu kết quả chính xác tuyệt đối.

+ Thực tế:

* **Quản lý lưu trữ**: Khi lưu trữ dữ liệu trên ổ cứng với dung lượng hạn chế, cần chọn tệp tin để tối ưu hóa giá trị sử dụng.
* **Hoạch định sản xuất**: Một nhà máy cần chọn máy móc hoặc nguyên liệu với giới hạn ngân sách để tối ưu hóa sản lượng.

### 3.3.2 Liên hệ thực tế

- Tham lam - Lựa chọn nhanh nhưng không đảm bảo tối ưu:

- Trong cuộc sống, giải thuật Tham lam giống như cách chúng ta đi siêu thị với một ngân sách cố định:

+ **Mục tiêu**: Tối đa hóa số lượng thực phẩm mua được.

+ **Cách làm**: Chọn các mặt hàng có giá trị dinh dưỡng cao nhất trên từng đồng chi phí.

+ **Hạn chế**: Có thể bỏ lỡ cơ hội tốt hơn khi kết hợp các mặt hàng khác với giá rẻ hơn nhưng tổng giá trị cao hơn.

- Quy hoạch động - Tìm kiếm tối ưu toàn cục:

- Tưởng tượng bạn lên kế hoạch cho một chuyến đi xa:

+ **Mục tiêu**: Chọn hành lý tối ưu nhất để mang theo trong vali có giới hạn cân nặng.

+ **Cách làm**: So sánh tất cả tổ hợp có thể của các món đồ (quần áo, sách, thiết bị điện tử, v.v.), sau đó chọn phương án tốt nhất.

+ **Kết quả**: Đảm bảo mang đủ đồ cần thiết mà không bị vượt cân.

- So sánh thực tế giữa hai giải thuật

|  |  |  |
| --- | --- | --- |
| **Trượng hợp thực tế** | **Giải thuật tham lam** | **Giải thuật quy hoạch động** |
| Mua sắm với ngân sách hạn chế | Chọn ngay các món đồ đang giảm giá mạnh nhất trước, không quan tâm tổng giá trị. | Xem xét tất cả các món đồ để tìm ra tổ hợp có tổng giá trị cao nhất trong ngân sách.. |
| Đóng gói đồ khi đi du lịch | Chọn các món đồ nhỏ và nhẹ nhất đầu tiên để tận dụng tối đa không gian vali. | Lên danh sách chi tiết, tính toán để mang được những món đồ cần thiết nhất và phù hợp với trọng lượng cho phép. |
| Quản lý bữa ăn hàng ngày | Ưu tiên món ăn có giá trị dinh dưỡng cao nhất mỗi bữa, không xét đủ nhu cầu tổng thể cả tuần. | Lập kế hoạch dinh dưỡng cả tuần, cân đối đủ protein, chất xơ, và năng lượng cho mọi bữa ăn. |
| Lựa chọn môn học trong kỳ học | Chọn các môn học có tín chỉ cao nhất hoặc dễ hoàn thành nhanh mà không quan tâm mục tiêu tốt nghiệp toàn diện. | Xem xét tổng thể các môn học để tối ưu tín chỉ và phù hợp với yêu cầu tốt nghiệp trong thời gian ngắn nhất. |
| Lựa chọn chương trình khuyến mãi | Chọn ngay chương trình có phần thưởng lớn nhất, không quan tâm các điều kiện nhỏ hoặc chi phí khác đi kèm. | So sánh kỹ các chương trình để tìm ra phần thưởng tốt nhất sau khi đã trừ đi tất cả chi phí và rủi ro liên quan. |
| Chơi trò chơi tích điểm | Tích điểm bằng cách chơi nhanh và đạt được số điểm cao nhất trong mỗi lượt chơi. | Lên kế hoạch chiến lược để tích lũy điểm tối đa qua nhiều lượt chơi, kể cả cần hy sinh một vài lượt để có điểm cao hơn. |

Bảng 3. 19 Bảng so sánh các trường hợp thực tế của hai giải thuật

### 3.3.3 Kết luận

- Giải thuật tham lam:

+ Tốc độ: Giải thuật Tham lam rất nhanh, đặc biệt khi cần xử lý lượng dữ liệu lớn.

+ Đơn giản: Dễ triển khai và áp dụng trong các tình huống yêu cầu tính toán nhanh mà không cần độ chính xác tuyệt đối.

+ Không luôn tối ưu: Vì chỉ tập trung vào lợi ích cục bộ (từng bước), nó có thể bỏ qua các tổ hợp toàn cục tốt hơn.

+ Khi tình huốn cần tính toán nhanh thì chúng ta nên sữ dụng giải thuật tham lam.

- Giải thuật quy hoạch động:

+ Tối ưu toàn cục: Đảm bảo tìm ra lời giải chính xác nhất cho mọi trường hợp.

+ Đa dạng bài toán: Thích hợp cho cả balo nguyên (không chia nhỏ đồ vật) và các bài toán tương tự khác.

+ Chậm hơn Tham lam: Đặc biệt khi số lượng đồ vật lớn và trọng lượng balo rất lớn.

+ Phức tạp hơn: Việc triển khai đòi hỏi phải hiểu rõ quy hoạch động và cách tối ưu hóa bộ nhớ (nếu cần).

+ Khi bài toán cần đua ra yêu cầu chính xác tuyệt đổi và không chấp nhận một giải pháp gần đúng thì khi đó giải thuật quy hoạch động là một lựa chọn tốt.

### 3.3.4 Hướng phát triển

Mở rộng bài toán cơ bản

- Balo phân số (Fractional Knapsack): Cho phép lấy một phần của một món đồ, thay vì phải chọn toàn bộ món đồ.

- Balo 0/1 (0/1 Knapsack): Mỗi món đồ chỉ có thể chọn hoặc không chọn, không thể chia nhỏ.

- Balo đa chiều (Multi-dimensional Knapsack): Thêm các ràng buộc khác ngoài trọng lượng, chẳng hạn như thể tích hoặc giá trị tối thiểu.

Nâng cao thuật toán

- Tối ưu hóa thuật toán:

+ Sử dụng lập trình động để giảm độ phức tạp.

+ Áp dụng thuật toán chia để trị.

+ Thuật toán heuristic hoặc metaheuristic, như Genetic Algorithm, Simulated Annealing, hoặc Ant Colony Optimization.

- Phân tích hiệu suất: Đo lường và tối ưu hóa hiệu suất, đặc biệt trong các bài toán với số lượng món đồ và giới hạn trọng lượng lớn.

Phát triển hệ thống hỗ trợ

- Xây dựng giao diện đồ họa: Giúp người dùng nhập dữ liệu và xem kết quả trực quan hơn.

- Thêm tính năng báo cáo: Hiển thị các lựa chọn tối ưu cùng phân tích chi tiết.

Mở rộng phạm vi nghiên cứu

- Balo ngẫu nhiên: Mô phỏng bài toán với các dữ liệu ngẫu nhiên.

- Balo động (Dynamic Knapsack): Giá trị hoặc trọng lượng món đồ thay đổi theo thời gian.

- Balo có ràng buộc phụ thuộc: Các món đồ có thể phụ thuộc vào nhau, ví dụ chọn món này thì không thể chọn món kia.

Áp dụng công nghệ hiện đại

- Machine Learning: Học cách dự đoán lựa chọn tối ưu từ các dữ liệu bài toán trước đó.

- Parallel Computing: Sử dụng tính toán song song để tăng tốc xử lý bài toán lớn.

### 3.3.5 Ưu điểm, hạn chế, nguyên nhân

- Ưu điểm:

+ Chương trình vận hành ổn đình.

+ Giao diện trực quan rõ ràng dễ sữ dụng.

+ Kết quả cho ra chính xác.

- Hạn chế:

+ Hiệu xuất chương trình sẻ chậm nếu người dùng nhập vào số món đồ và trong lớn balo lớn.

+ Giao diện nhập thủ công yêu cầu người dùng phải nhập theo quy định cụ thể đôi lúc sẻ cảm thấy bất tiện.

+ Thiết kế thuật toán chưa được tối ưu qua nhiều vòng lặp đôi lúc dẫn đến treo chương trình.

- Nguyên nhân:

+ Do kiến thức còn hạn chế nên việc thiết kế thuật toán chưa được tối ưu.

+ Thuật toán sử dụng có độ phức tạp, dẫn đến việc tính toán tốn nhiều thời gian khi dữ liệu đầu vào lớn.

+ Thiếu kinh nghiệm trong việc thiết kế giao diện người dùng, chưa tích hợp các tính năng thân thiện như kiểm tra lỗi tự động hoặc nhập dữ liệu từ file.

+ Do chưa tìm hiểu sâu về các công nghệ hiện đại hỗ trợ giao diện nhập liệu tiện lợi, dẫn đến giao diện chưa đáp ứng tốt trải nghiệm người dùng.

# TÀI LIỆU THAM KHẢO

🙞 🙢 🕮 🙠 🙜

**Tài liệu tiếng việt**

[1]. Lê Thanh Trúc. Phân tích và thiết kế thuật toán. Khoa Kỹ Thuật – Công Nghệ, Đại Học Tây Đô, 2023.

[2]. Nguyễn Chí Cường. Cấu trúc dữ liệu & Giải thuật, Chương 5. Khoa Kỹ thuật – Công nghệ, Đại Học Tây Đô, 2023.

[3]. Nguyễn Chí Cường. Lập trình căn bản. Khoa Kỹ thuật – Công nghệ, Đại Học Tây Đô, 2023.

[4]. Lâm Tấn Phương. Giáo trình Lý thuyết Thiết kế và lập trình Web, Khoa Kỹ thuật – Công nghệ, Đại Học Tây Đô, 2024.

**Tài liệu từ Internet**

1. Hoang Minh Dai. Thuật toán QUY HOẠCH ĐỘNG <https://viblo.asia/p/bai-toan-cai-tui-va-nhung-ung-dung-xung-quanh-no-maGK7Nke5j2> (12/2024)
2. Trần Trung Phong. Thuật toán tham lam - Greedy Algorithm <https://viblo.asia/p/thuat-toan-tham-lam-greedy-algorithm-XQZGxozlvwA> (12/2024)