**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**

**Ngành: Công nghệ Thông Tin**

**Sinh viên thực hiện: Nguyễn Chí Nguyện – MSSV: 227060172**

**Cần Thơ – tháng 12 năm 2024**

**LỜI CẢM ƠN**

Để hoàn thành niên luận này, chúng em xin gửi lời cảm ơn chân thành đến:

Ban giám hiệu trường Đại Học Tây Đô vì đã tạo điều kiện về cơ sở vật chất với hệ thống thư viện hiện đại, đa dạng các loại sách, tài liệu thuận lợi cho việc tìm kiếm, nghiên cứu thông tin.

Xin cảm ơn giảng viên thạc sĩ – **Thầy Nguyễn Chí Cường** đã giảng dạy tận tình, chi tiết để chúng em có đủ kiến thức và vận dụng chúng vào bài niên luận này.

Do chưa có nhiều kinh nghiệm làm đề tài cũng như những hạn chế về kiến thức, trong bài niên luận chắc chắn sẽ không tránh khỏi những thiếu sót. Rất mong nhận được sự nhận xét, ý kiến đóng góp, phê bình từ phía cô để bài niên luận được hoàn thiện hơn.

Lời cuối cùng, em xin kính chúc thầy nhiều sức khỏe, thành công và hạnh phúc. Chúng em xin chân thành cảm ơn.

**SINH VIÊN THỰC HIỆN**

**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*

**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)* |  |
| * Đặc tả hệ thống (0.5 điểm) * Các mô hình (MCD, MLD, PDM) và các ràng buộc toàn vẹn (1 điểm) * DFD, Sơ đồ chức năng, lưu đồ giải thuật giải quyết vấn đề (nếu có), diễn giải ý nghĩa các màn hình cập nhật (FORM) và các báo biểu (REPORT) (1 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, cài đặt CSDL cùng với các ràng buộc toàn vẹn, Forms cập nhật đúng, Reports hợp lý… *(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** |
| **………………..** |

**CHƯƠNG 1: TỔNG QUAN**

🖎🕮✍

**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:

1. 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.
2. 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.

**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 vật dụng cần thiết, mỗi món đồ có giá trị và trọng lượng riêng: Đồ ăn: Giá trị là 8, trọng lượng 5kg; Lều: Giá trị là 10, trọng lượng 7kg; Bộ dụng cụ y tế: Giá trị là 6, trọng lượng 4kg; Nước uống: Giá trị là 5, trọng lượng 3kg; Đèn pin: Giá trị là 4, trọng lượng 2kg; Áo ấm: Giá trị là 7, trọng lượng 6kg.

**3. MỤC TIÊU CẦN ĐẠT ĐƯỢC**

**Dữ liệu đầu vào**:

Nhập danh sách đồ vật với trọng lượng và giá trị tương ứng từ bàn phím.

Nhập trọng lượng tối đa của balo.

**Đầu ra mong muốn**:

Tìm giá trị lớn nhất có thể chứa được trong balo.

Xuất danh sách các đồ vật được chọn với tổng trọng lượng không vượt quá giới hạn bằng hai giải thuật rôid so sánh với nhau.

**4. HƯỚNG GIẢI QUYẾT**

**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**.

**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.

**5. KẾ HOẠCH THỰC HIỆN**

|  |  |
| --- | --- |
| **Thời gian** | **Công việc** |
| **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**

🙞 🙢 🕮 🙠 🙜

**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. CSS (Cascading Style Sheets)**

- 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.

**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.

**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.

**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.

**6. VẬN DỤNG LÝ THUYẾT VÀO BÀI TẬP**

**6.1 Giải thuật quy hoạch động**

Có 4 đồ vật, trọng lượng tối đa 10kg

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

**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 |  |  |  |  |  |  |  |  |  |

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 |  |  |  |  |  |  |  |  |

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 |  |  |  |  |  |  |  |

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 | 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 |  |  |  |  |  |

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[5], 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 |  |  |  |  |  |

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 |  |  |  |  |

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 |  |  |  |

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 |  |  |

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 |  |

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 |

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 |

- 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:**

Chọn được: đồ vật 1 chọn 1 lận, đồ vật 2 chọn 3 lần

Giá trị lớn nhất: 13

Đơn giá: 5$

Trọng lượng còn lại: 0

**6.2 Lưu đồ giải thuật**

|  |
| --- |
| Đúng |

|  |
| --- |
| Đúng |

|  |
| --- |
| Đúng |

|  |
| --- |
| Đúng |

|  |
| --- |
| Sai |

|  |
| --- |
| Sai |

|  |
| --- |
| Sai |

j++

Cập nhật GTLN cột j = GTLN của cột [cột j – trọng lượng vật i] + giá trị vật i Lưu lại lần chọn j = i

GTLN cột j < GTLN của cột [cột j – trọng lượng vật i] + giá trị vật i

W[i] <= trọng lượng j

Khởi tạo j = 0

j<=trọng lượng balo

i++

i<=số món đồ

Khởi tạo i = 0

Giải thuật quy hoạch động

Hoàn thành

|  |
| --- |
| Sai |

**6.3 Giải thuật tham lam**

Có 4 đồ vật, trọng lượng tối đa 10kg

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

Bước 1: Sắp xếp các đồ vật theo giá trị giảm dần

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

Bước 2:

Đồ vật 4

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

0 + 6 = 6 <= 10

Chọn đồ vật 1

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

6 + 6 = 12 <= 10 => Dừng

Đồ vật 3

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

10 + 3 = 13 <= 10

Dừng

Đồ vật 1

10 + 1 = 11 <= 10 => Dừng

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  
**Tổng đơn giá**: 2.58$

**6**.**4 Lưu đồ giải thuật tham lam**

i++

Hoàn thành

Tổng giá trị += Giá trị đồ vật i, Tổng trọng lượng += Trọng lượng đồ vật i, Tăng số lượng món đồ được chọn, Tăng số lần chịn đồ vật i

Tổng trọng lượng += trọng lượng đồ vật i < Trọng lượng balo

i<Độ dài Arr

Khởi tạo Tổng giá trị = 0, Tổng đơn giá = 0, Lần chọn các đồ vật = 0, i = 0

Giải thuật tham ăn

|  |
| --- |
| Sai |

|  |
| --- |
| Đúng |

|  |
| --- |
| Sai |

|  |
| --- |
| Đúng |

**CHƯƠNG 3: KẾT QUẢ ỨNG DỤNG**

🙞 🙢 🕮 🙠 🙜

**1. 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

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.









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

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

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

A screenshot of a cell phone

Description automatically generatedA screenshot of a computer

Description automatically generated

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

A white rectangular object with a blue border

Description automatically generated

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

A screenshot of a video game

Description automatically generated

A screenshot of a calendar

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

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

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

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

+ : là giá trị của đồ vật thứ (i).

+ j : là mỏi kg của trọng lượng balo.

+ 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

Bên phải là đáp án của giải thuật quy hoạch động bên trái là đáp án của giải thuật tham lam.

**CHƯƠNG 4: KẾT LUẬN**

🙞 🙢 🕮 🙠 🙜

**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. |

- 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.

**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. |

**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.

**CHƯƠNG 6: TÀI LIỆU THAM KHẢO**

🙞 🙢 🕮 🙠 🙜

1. <https://www.w3schools.com/>

2. https://vi.wikipedia.org/wiki/Bài\_toán\_xếp\_ba\_lô

3. https://www.youtube.com/watch?v=8JQOOiFVCMk