**Tham ăn**

Tham ăn (Greedy algorithm) là một kỹ thuật được dùng nhiều để giải các bài toán tối ưu tổ hợp bằng cách tìm kiếm lựa chọn tối ưu ở mỗi bước đi với hy vọng tìm được phương án tối ưu toàn cục để xây dựng một phương án.

**Nội dung:**

-Kỹ thuật tham ăn thường được vận dụng để giải bài toán tối ưu tổ hợp bằng cách xây dựng một phương án X.

-Phương án X được xây dựng bằng cách lựa chọn từng thành phần Xi của X cho đến khi hoàn chỉnh (đủ n thành phần).

-Với mỗi Xi, ta sẽ chọn Xi tối ưu. Với cách này thì có thể ở bước cuối cùng ta không còn gì để chọn mà phải chấp nhận một giá trị cuối cùng còn lại.

-Áp dụng kỹ thuật tham ăn sẽ cho một giải thuật thời gian đa thức, tuy nhiên nói chung chúng ta chỉ đạt được một phương án tốt chứ chưa hẳn là tối ưu.

**Mô tả:**

Bước 1 :Tính đơn giá cho các loại đồ vật.

Bước 2 :Xét các loại đồ vật theo thứ tự đơn giá từ lớn đến nhỏ.

Bước 3 :Với mỗi đồ vật được xét sẽ lấy một số lượng tối đa mà trọng lượng còn lại của ba lô cho phép.

Bước 4 :Xác định trọng luợng còn lại của ba lô và quay lại bước 3 cho đến khi không còn có thể chọn được đồ vật nào nữa.

**Nhánh cận**

Nhánh cận (Branch and bound) là kỹ thuật xây dựng cây tìm kiếm phương án tối ưu, nhưng không xây dựng toàn bộ cây mà sử dụng một giá trị cận để cắt bỏ bớt một số nhánh.

**Nội dung:**

-Cây tìm kiếm phương án có nút gốc biểu diễn cho tập tất cả các phương án có thể có, mỗi nút lá biểu diễn cho một phương án nào đó. Nút n có các nút con tương ứng với các khả năng có thể lựa chọn tập phương án xuất phát từ n. Kỹ thuật này gọi là **phân nhánh**.

-Với mỗi nút trên cây ta sẽ xác định một giá trị cận. Giá trị cận là một giá trị gần với giá của các phương án. Với bài toán tìm min ta sẽ xác định cận dưới còn với bài toán tìm max ta sẽ xác định cận trên. Cận dưới là giá trị nhỏ hơn hoặc bằng giá của phương án, ngược lại cận trên là giá trị lớn hơn hoặc bằng giá của phương án.

-Đối với thuật toán nhánh cận, trong quá trình đi tìm lời giải của một phương án, thuật toán sẽ tính điều kiện dựa vào một giá trị cận để xác định những phương án không tối ưu mà loại bỏ sớm các phương án đó, từ đó sẽ quyết định đi theo nhánh có cận cho là tốt nhất để tìm được phương án tối ưu nhất

**Mô tả :**

Bước 1 :Danh sách các đồ vật được sắp xếp theo thứ tự giảm của đơn giá.

Bước 2 :Nút gốc biểu diễn cho trạng thái ban đầu của ba lô, ở đó ta chưa chọn một vật nào.Tổng giá trị được chọn TGT = 0. Cận trên của nút gốc CT = W \* Ðơn giá lớn nhất.

Nút gốc sẽ có các nút con tương ứng với các khả năng chọn đồ vật có đơn giá lớn nhất. Với mỗi nút con ta tính lại các thông số:

-TGT = TGT (của nút cha) + số đồ vật được chọn \* giá trị mỗi vật.

-W  = W (của nút cha) - số đồ vật được chọn \* trọng lượng mỗi vật.

-CT = TGT + W \* Ðơn giá của vật sẽ xét kế tiếp.

Bước 3 : Trong các nút con, ta sẽ ưu tiên phân nhánh cho nút con nào có cận trên lớn hơn trước. Các con của nút này tương ứng với các khả năng chọn đồ vật có đơn giá lớn tiếp theo. Với mỗi nút ta lại phải xác định lại các thông số TGT, W, CT theo công thức đã nói trong bước 2.

Lặp lại bước 3 với chú ý: đối với những nút có cận trên nhỏ hơn hoặc bằng giá lớn nhất tạm thời của một phương án đã được tìm thấy thì ta không cần phân nhánh cho nút đó nữa (cắt bỏ).

Nếu tất cả các nút đều đã được phân nhánh hoặc bị cắt bỏ thì phương án có giá lớn nhất là phương án cần tìm.

**Quy hoạch động**

Quy hoạch động là một kỹ thuật giúp đơn giản hóa việc tính toán các công thức truy hồi bằng cách lưu trữ giá trị kết quả của các bài toán con vào các ô của bảng dữ liệu với  mục đích sử dụng lại giá trị khi cần.

**Nội dung :**

Trong giải thuật đệ quy, một số bài toán con nào đó bị giải nhiều lần.

Tạo ra một bảng để lưu trữ kết quả của các bài toán con và khi cần chúng ta sẽ sử dụng kết quả đã được lưu trong bảng mà không cần phải giải lại bài toán đó.

Tạo bảng bằng cách:

* Gán giá trị cho một số ô nào đó.
* Gán trị cho các ô khác nhờ vào giá trị của các ô trước đó.

Tra bảng và xác định kết quả của bài toán ban đầu.

Mô tả:

Giả sử X[k,V] là số lượng đồ vật k được chọn, F[k,V] là tổng giá trị của k đồ vật đã được chọn và V là trọng lượng còn lại của ba lô, k = 1..n, V = 0..W.

Trong trường hợp đơn giản nhất, khi chỉ có một đồ vật, ta tính được X[1,V] và F[1,V] với mọi V từ 0 đến W như sau: X[1,V] = V/g1 và F[1,V] = X[1,V] \* v1.

Giả sử ta đã tính được F[k-1,V], khi có thêm đồ vật thứ k, ta sẽ tính được F[k,V], với mọi V từ 0 đến W. Cách tính như sau: Nếu ta chọn xk đồ vật loại k, thì trọng lượng còn lại của ba lô dành cho k-1 đồ vật từ 1 đến k-1 là U = V-xk\*gk và tổng giá trị của k loại đồ vật đã được chọn F[k,V] = F[k-1,U] + xk\*vk, với xk thay đổi từ 0 đến yk= V/gk và ta sẽ chọn xk sao cho F[k,V] lớn nhất.

Ta có công thức truy hồi như sau:

X[1,V] = V/g1 và F[1,V] = X[1,V] \* v1.

F[k,V] = Max(F[k-1,V-xk\*gk] + xk\*vk) với xk chạy từ 0 đến V/gk.

Sau khi xác định được F[k,V] thì X[k,V] là xk ứng với giá trị F[k,V] được chọn trong công thức trên.

Để lưu các giá trị trung gian trong quá trình tính F[k,V] theo công thức truy hồi trên, ta sử dụng một bảng gồm n dòng từ 1 đến n, dòng thứ k ứng với đồ vật loại k và W+1 cột từ 0 đến W, cột thứ V ứng với trọng lượng V. Mỗi cột V bao gồm hai cột nhỏ, cột bên trái lưu F[k,V], cột bên phải lưu X[k,V]. Trong lập trình ta sẽ tổ chức hai bảng tách rời là F và X.

**CƠ SỞ DỮ LIỆU**

Sử dung cơ sở dữ liệu MySQL.

MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở, rất phổ biến, có tốc độ truy cập và bảo mật cao, ổn định và dễ sử dụng được dùng để quản lý hệ thống cơ sở dữ liệu.

**Bảng balo**

Bảng mô tả balo: Chứa thông tin của balo: id, mso, khoiluong, loai, soluongdv.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | id | int | Được tăng tự động và là khóa chính để phân biệt với những balo khác trong cơ sở dữ liệu. |
| 2 | mso | varchar | Mã số của balo, có khóa UNIQUE để phân biệt với các mã số balo khác trong cơ sở dữ liệu |
| 3 | khoiluong | float | Tổng khối lượng balo có thể đựng được |
| 4 | loai | int | Loại của balo |
| 5 | soluongdv | int | Số lượng loại đồ vật của balo |

**Bảng dovat**

Bảng mô tả đồ vật: Chứa thông tin của đồ vật: id, stt, tendovat, khoiluong, giatri, soluong, idbalo.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | id | int | Được tăng tự động và là khóa chính để phân biệt với những đồ vật khác trong cơ sở dữ liệu. |
| 2 | idbalo | int | id của balo mà đồ vật thuộc |
| 3 | tendovat | varchar | Tên của đồ vật |
| 4 | khoiluong | float | Khối lượng của đồ vật |
| 5 | giatri | float | Giá trị của đồ vật |
| 6 | soluong | int | Số lượng của loại đồ vật |