**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**





**ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN**

**TÊN ĐỀ TÀI**: BÀI TOÁN VẬN TẢI

Người hướng dẫn**: TS. Đặng Thiên Bình**

Sinh viên thực hiện**:**

**NGUYỄN VĂN BẮC MSV: 102220005 NHÓM: 8**

**LÊ VĂN QUANG MSV: 102220035 NHÓM: 8**

**Đà Nẵng, 06/2023**

MỤC LỤC

[DANH MỤC HÌNH VẼ](#_Toc130626351)

[MỞ ĐẦU](#_Toc130626352)

[1. TỔNG QUAN ĐỀ TÀI](#_Toc130626353)

[2. CƠ SỞ LÝ THUYẾT](#_Toc130626354)

[2.1. Ý tưởng](#_Toc130626355)

[2.2. Cơ sở lý thuyết](#_Toc130626356)

[3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN](#_Toc130626357)

[3.1. Phát biểu bài toán](#_Toc130626358)

[3.2. Cấu trúc dữ liệu](#_Toc130626359)

[3.3. Thuật toán](#_Toc130626360)

[4. CHƯƠNG TRÌNH VÀ KẾT QUẢ](#_Toc130626361)

[4.1. Tổ chức chương trình](#_Toc130626362)

[4.2. Ngôn ngữ cài đặt](#_Toc130626363)

[4.3. Kết quả](#_Toc130626364)

[4.3.1. Giao diện chính của chương trình](#_Toc130626365)

[4.3.2. Kết quả thực thi của chương trình](#_Toc130626366)

[4.3.3. Nhận xét đánh giá](#_Toc130626367)

[5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN](#_Toc130626368)

[5.1. Kết luận](#_Toc130626369)

[5.2. Hướng phát triển](#_Toc130626370)

[TÀI LIỆU THAM KHẢO](#_Toc130626371)

[PHỤ LỤC](#_Toc130626372)

DANH MỤC HÌNH VẼ

Hình 1: Xây dựng bảng chi phí vận tải vận tải .

Hình 2: Thuật toán leastcost xử lý bảng vận tải.

Hình 3,4: Giao diện chính của chương trình

MỞ ĐẦU

**\*Mục đích thực hiện đề tài: thông qua quá trình thực hiện đề tài, ta có thêm kiến thức cũng như áp dụng những kiến thức đã học và nghiên cứu được để giải quyết vấn đề, bài toán đặt ra một cách tối ưu nhất có thể.**

**\*Mục tiêu: giải quyết bài toán tối ưu nhất có thể.**

**\*Phạm vi và đối tượng nghiên cứu: sử dụng các kiến thức đã học và nguồn tài liệu mở. Đối tượng nghiên cứu chính là các phương pháp tối ưu cho bài toán và cấu trúc dữ liệu với độ phức tạp thấp.**

**\*Phương pháp nghiên cứu: sử dụng các kiến thức đã học và tài liệu tìm kiếm được trên Internet, tìm hiểu và đưa ra cấu trúc dữ liệu phù hợp từ đó hoàn thành được bài toán.**

**\*Cấu trúc của đồ môn học: sau khi tìm hiểu về các lý thuyết của bài toán và các lý thuyết liên quan, lên ý tưởng giải quyết vấn đề, kiểm tra và nâng cấp thuật toán.**

# TỔNG QUAN ĐỀ TÀI

Từ lâu ngành giao thông vận tải đã trở thành một ngành có vai trò rất quan trọng trong sự phát triển kinh tế toàn cầu, đảm bảo quá trình sản xuất, giao thương diễn ra liên tục,ổn định,đáp ứng nhu cầu di chuyển của con người…vv.

Đối với ngành giao thông vận tải việc thiết lập cách thức vận chuyển,tính toán chi phí vận chuyển hợp lý là vô cùng quan trọng và thiết yếu,nó đã trở thành vấn đề cơ bản của ngành vận tải gọi là ***bài toán vận tải.***

Bài toán đặt ra vấn đề là làm thế nào để phân phối một lượng hàng hóa từ các nguồn cung ứng đến các điểm tiêu dùng một cách hiệu quả nhất, sao cho chi phí vận chuyển là thấp nhất có thể. Bài toán có nhiều ứng dụng thực tiễn trong các lĩnh vực như quản lý chuỗi cung ứng, giao thông vận tải, sản xuất công nghiệp, v.v. Bài toán vận tải cũng liên quan đến nhiều lĩnh vực khoa học khác như tối ưu hóa toán học, lý thuyết trò chơi, lý thuyết đồ thị, v.v.

Với sự phát triển của nghành công nghệ thông tin, việc nghiên cứu cũng như tính toán đã trở nên dễ dàng hơn, chỉ mất vài giây hoặc thậm chí ít hơn để máy tính tính toán được đường đi tối ưu nhất theo nhiều phương pháp.

Đề tài nhằm xây dựng chương trình chứa thuật toán để tìm dường đi và chi phí nhỏ nhất cho bài toán vận tải.với các dữ liệu nhập vào từ người sử dụng.

# CƠ SỞ LÝ THUYẾT

## Ý tưởng

-Giả thiết có m kho hàng chứa lượng hàng A1,A2,A3,..Am cung cấp hàng cho n cửa hàng với lượng hàng cần được cung cấp là B1,B2,B3,..,Bn

-Gọi Cij,Xij lần lượt là lượng hàng và giá thành vận chuyển mỗi đơn vị hàng từ kho hàng Ai đến cửa hàng Bj ,khi đó tổng giá thành vận chuyển là

f(x)= .đây là một phương trình tuyến tính do đó để tìm f(x)min ta sẽ giải bài toán bằng quy hoạch tuyến tính, trước tiên ta thiết lập bài toán dưới dạng bảng:

A picture containing text, screenshot, square, line

Description automatically generatedHình 1.

## Cơ sở lý thuyết

2.2.1. Quy hoạch tuyến tính(Linear programming):

Quy hoạch tuyến tính là một phương pháp toán học để giải quyết các bài toán tối ưu hóa trong đó các ràng buộc và hàm mục tiêu đều là các hàm tuyến tính. Quy hoạch tuyến tính có thể được sử dụng để giải quyết nhiều loại bài toán khác nhau, bao gồm cả bài toán vận tải.

Để giải quyết một bài toán quy hoạch tuyến tính, chúng ta cần xác định các biến quyết định, các ràng buộc và hàm mục tiêu. Các biến quyết định là các giá trị mà chúng ta cần tìm để tối ưu hóa hàm mục tiêu. Các ràng buộc là các điều kiện mà các biến quyết định phải thỏa mãn. Hàm mục tiêu là một hàm tuyến tính của các biến quyết định mà chúng ta muốn tối ưu hóa.

2.2.2. Phương pháp cước phí nhỏ nhất:

Phương pháp cước phí nhỏ nhất là một phương pháp heuristic được sử dụng để giải quyết bài toán vận tải. Đây là một phương pháp đơn giản và hiệu quả để tìm ra một phương án cơ bản ban đầu cho bài toán.

Phương pháp cước phí nhỏ nhất là một cách đơn giản và hiệu quả để tìm ra một giải pháp khả thi cho bài toán vận tải. Tuy nhiên, nó chỉ cung cấp một giải pháp xấp xỉ và có thể không đảm bảo tìm ra được giải pháp tối ưu. Do đó, chúng ta có thể sử dụng nó kết hợp với các thuật toán khác để tìm ra giải pháp tốt hơn.

# TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

## Phát biểu bài toán

Mô tả đầu vào (Input) và đầu ra (Output)

Input:

* Một ma trận chi phí cost kích thước m x n, trong đó cost[i][j] là chi phí vận chuyển hàng hóa từ điểm cung cấp thứ i đến điểm nhận thứ j.
* Một mảng supply kích thước m, trong đó supply[i] là lượng hàng hóa cung cấp tại điểm cung cấp thứ i.
* Một mảng demand kích thước n, trong đó demand[j] là lượng hàng hóa yêu cầu tại điểm nhận thứ j.

Output: đường đi với tổng cước phí nhỏ nhất.

## Cấu trúc dữ liệu

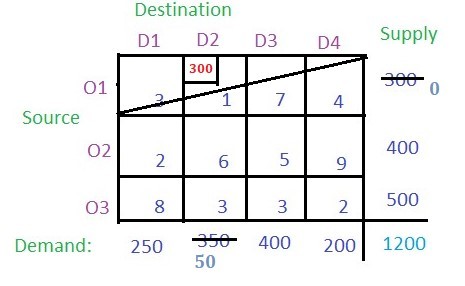
-Mảng( ma trận ): Trong khoa học máy tính, mảng là một cấu trúc dữ liệu bao gồm một nhóm các phần tử giá trị hoặc biến, mỗi phần tử được xác định ít nhất bằng một chỉ số (index) hoặc khóa (key). Mảng được lưu theo cách có thể tính được vị trí của các phần tử từ giá trị của một bộ (tuple) chỉ số bằng một biểu thức toán học. Mảng là một trong những cấu trúc dữ liệu cũ và quan trọng nhất, và hầu hết các chương trình đều dùng nó. Các cấu trúc dữ liệu khác cũng được hiện thực bằng mảng, thí dụ như danh sách hoặc chuỗi hoặc hiện thực các bảng, nhất là bảng tìm kiếm.

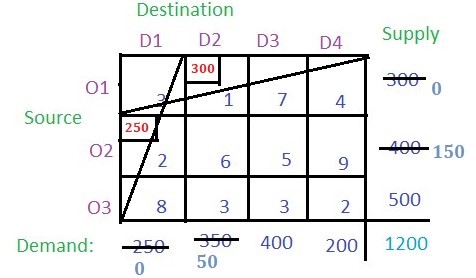
-Struct: Struct là một kiểu dữ liệu tự định nghĩa trong ngôn ngữ lập trình C. Nó cho phép bạn lưu trữ các biến có kiểu dữ liệu khác nhau dưới dạng một nhóm dữ liệu duy nhất. Các biến này được gọi là các thành viên của struct. Struct thường được sử dụng để đóng gói các biến liên quan với nhau thành một đơn vị dữ liệu hợp lý.

-Heap: Heap là một dạng cấu trúc dữ liệu cây nhị phân đặc biệt, trong đó các nút con của một nút luôn nhỏ hơn (hoặc lớn hơn) nút cha của chúng. Có hai loại heap phổ biến là min heap và max heap. Trong min heap, giá trị của mỗi nút luôn nhỏ hơn hoặc bằng giá trị của các nút con của nó, trong khi đó trong max heap, giá trị của mỗi nút luôn lớn hơn hoặc bằng giá trị của các nút con của nó.

**3.3. Thuật toán**

Đầu tiên ta tìm kiếm ô có chi phí thấp nhất trong bảng chi phí. Sau đó, ta cấp phát cho ô đó lượng hàng hóa lớn nhất có thể, tức là lượng hàng hóa nhỏ hơn trong hai giá trị: lượng hàng hóa cung cấp tại điểm cung cấp tương ứng và lượng hàng hóa yêu cầu tại điểm nhận tương ứng. Sau khi cấp phát xong, ta gạch ngang dòng hoặc cột tương ứng với điểm cung cấp hoặc điểm nhận đã được đáp ứng đủ nhu cầu và tiếp tục tìm kiếm ô có chi phí thấp nhất trong các ô còn lại. Quá trình này được lặp lại cho đến khi tất cả hàng hóa đều được vận chuyển hoặc tất cả nhu cầu đều được đáp ứng.



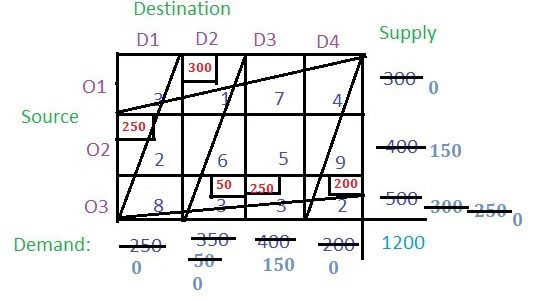


A picture containing diagram, line, screenshot, plot

Description automatically generated

A picture containing diagram, line, screenshot, text

Description automatically generated



A picture containing diagram, line, text, screenshot

Description automatically generated

1. CHƯƠNG TRÌNH VÀ KẾT QUẢ

## Tổ chức chương trình

* + 1. Xây dựng một cây dữ liệu MinHeap.
    2. Xây dựng hàm LeastCost.
    3. Xây dựng hàm main để thực thi bài toán cũng như hiển thị giao diện chương trình.

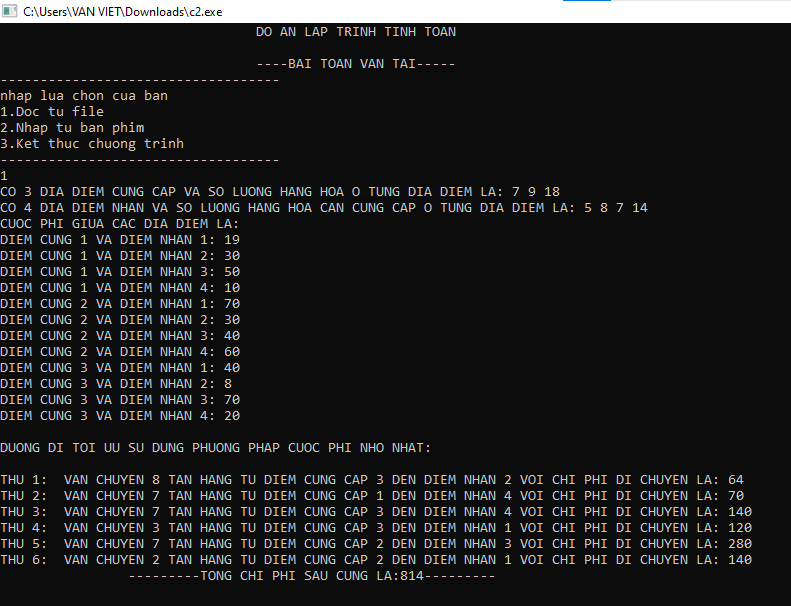
## Ngôn ngữ cài đặt

-Ngôn ngữ C: C là một ngôn ngữ mệnh lệnh được phát triển từ đầu thập niên 1970 bởi Dennis Ritchie để dùng trong hệ điều hành UNIX. Từ đó, ngôn ngữ này đã lan rộng ra nhiều hệ điều hành khác và trở thành một những ngôn ngữ phổ dụng nhất. C là ngôn ngữ rất có hiệu quả và được ưa chuộng nhất để viết các phần mềm hệ thống, mặc dù nó cũng được dùng cho việc viết các ứng dụng. Ngoài ra, C cũng thường được dùng làm phương tiện giảng dạy trong khoa học máy tính mặc dù ngôn ngữ này không được thiết kế dành cho người nhập môn.

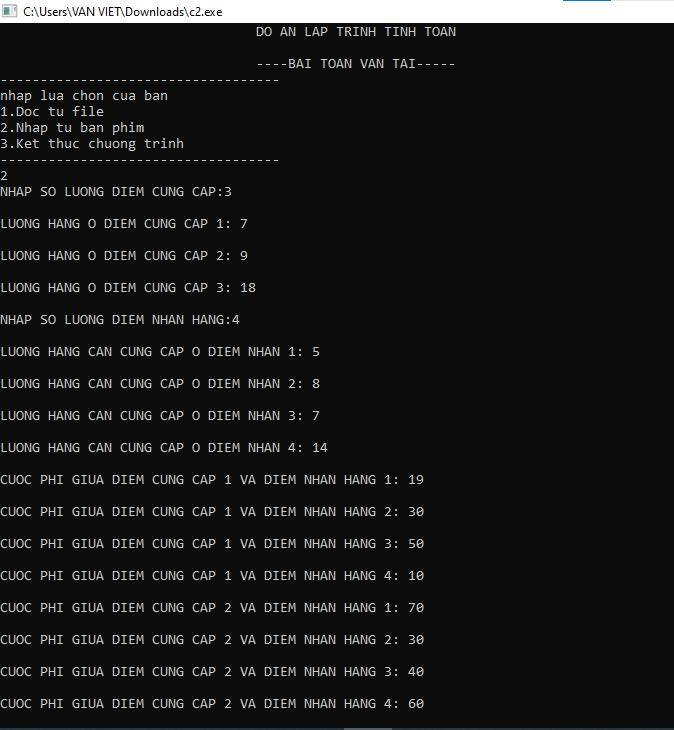
## Kết quả

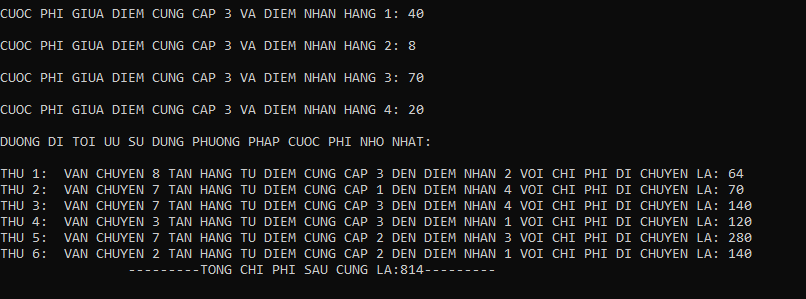
### Giao diện chính của chương trình

-Khi nhập dữ liệu từ file: Hình 3



-Khi nhập trực tiếp từ bàn phím: Hình 4





### Kết quả thực thi của chương trình

Hiển thị đường đi, lượng hàng hóa cần cấp phát, chi phí vận tải trên mỗi chuyến đi và tổng chi phí sau cùng.

### Nhận xét đánh giá

-Thuật toán sử dụng cây MinHeap và thuật toán tìm kiếm heapify kết hợp với mảng nên độ phức tạp thời gian tính toán được là O(m\*nlog(m\*n)) nên thời gian thực thi bài toán tương đối nhanh, đủ để áp dụng cho các bài tập trên lớp hoặc tình huống thực tế có bộ dữ liệu không quá lớn.

-Kết quả đưa ra đạt được độ chính xác như mong muốn.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

# 5.1 Trong đồ án lập trình này, bài toán vận tải được giải quyết bằng cách sử dụng phương pháp cước phí nhỏ nhất. Hàm xử lý chính leastCost được triển khai để giải quyết bài toán này và đã đạt được kết quả tốt. Hàm leastCost hoạt động hiệu quả và có thể giải quyết các bài toán vận tải với kích thước lớn một cách nhanh chóng.

**5.2 Hướng phát triển**

Tuy nhiên thuật toán LeastCost chỉ là giải pháp ban đầu, trong một vài trường hợp thì đây chưa phải là phương pháp tối ưu, cần nghiên cứu và phát triển một thuật toán tối ưu hơn cũng như các công cụ khác để người dung có thể tiếp cận bài toán dễ dàng hơn.

Mặt khác “bài toán vận tải” là bài toán cơ sở nên từ đó ta có thể phát triển thành nhiều bài toán có độ ứng dụng cao hơn như:

\*Bài toán vận tải đa mục tiêu:Giải quyết vấn đề cân bằng thu phát đồng thời tối ưu hóa một số mục tiêu khác như thời gian giao hàng,..v.v

\*Bài toán vận tải đa phương tiện:giải quyết các vấn đề của bài toán với các hình thức vận chuyển khác nhau như đường thủy,đường bộ, đường hàng không,.. với các phương tiện như oto,máy bay,tàu thuyền,..

\*Bài toán vận tải địa lý:xây dựng các mô hình vận tải bằng để giải quyết các yêu cầu địa lý đặc thù.

\*Bài toán vận tải thời gian thực:Áp dụng các giải pháp công nghệ mới như internet of Things và AI để tự động hóa việc điều phối và quản lý các phương tiện vận tải trong thời gian thực.

\*Bài toán vận tải năng lương:như khí đốt,..v.v

TÀI LIỆU THAM KHẢO

[Microsoft PowerPoint - chuong5\_THQL (hcmut.edu.vn)](http://www4.hcmut.edu.vn/~dotxlan/THQL/tailieumonhoc/chuong5_THQL.pdf)

[Quy hoạch tuyến tính – Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/Quy_ho%E1%BA%A1ch_tuy%E1%BA%BFn_t%C3%ADnh)

PHỤ LỤC

Sinh viên bỏ Code từng phần vào đây.{*Font: Time New Roman; thường; cỡ chữ: 12; dãn dòng: 1,3; căn lề: justified*

typedef struct node {

int i;

int j;

int cost;

}Node;

void swap(node \*a,node \*b)

{

node temp =\*a;

\*a=\*b;

\*b=temp;

}

void heapify(Node heap[], int n, int i) {

int smallest = i;

int left = 2 \* i + 1;

int right = 2 \* i + 2;

if (left < n && heap[left].cost < heap[smallest].cost) {

smallest = left;

}

if (right < n && heap[right].cost < heap[smallest].cost) {

smallest = right;

}

if (smallest != i) {

swap(&heap[i], &heap[smallest]);

heapify(heap, n, smallest);

}

}

void push(Node heap[], int \*n, Node newNode) {

if (\*n == INT\_MAX) {

return;

}

(\*n)++;

int i = \*n - 1;

heap[i] = newNode;

while (i != 0 && heap[(i - 1) / 2].cost > heap[i].cost) {

swap(&heap[(i - 1) / 2], &heap[i]);

i = (i - 1) / 2;

}

}

Node pop(Node heap[], int \*n) {

if (\*n == 0) {

return (Node){-1, -1, -1};

}

Node root = heap[0];

heap[0] = heap[\*n - 1];

(\*n)--;

heapify(heap, \*n, 0);

return root;

}

void leastCost(int \*\*cost, int \*supply, int \*demand, int m, int n) {

Node heap[m\*n];

int size = 0;

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

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

if (cost[i][j] > -1) {

push(heap, &size, (Node){i, j, cost[i][j]});

}

}

}

int totalCost = 0;

printf("DUONG DI TOI UU SU DUNG PHUONG PHAP CUOC PHI NHO NHAT:\n");

while (size > 0) {

Node tem = pop(heap, &size);

if (supply[temp.i] > 0 && demand[tem.j] > 0) {

int x = supply[tem.i] < demand[tem.j] ? supply[tem.i] : demand[tem.j];

totalCost += x \* tem.cost

supply[tem.i] -= x;

demand[tem.j] -= x

printf("x[%d][%d] = %d\n", tem.i + 1, tem.j + 1, x);

}

}

printf("Total cost: %d\n", totalCost);