**ĐỀ CƯƠNG BÀI TẬP LỚN**

**Tên đề tài:** Tìm hiểu bài toán phân công công việc và cài đặt thuật toán để giải bài toán

**Bước 1: Phân tích yêu cầu của bài toán**

1. **Bài toán phân công công việc**

* Mô tả: Bài toán phân công công việc là tìm cách phân bổ công việc cho các nhân viên sao cho tổng chi phí là nhỏ nhất, trong đó mỗi công việc chỉ được thực hiện bởi một nhân viên duy nhất.

1. **Yêu cầu của bài toán**

* Bài toán yêu cầu cài đặt một thuật toán để tìm phương án phân công công việc tối ưu.
* Đầu vào là một ma trận chi phí, trong đó mỗi ô chứa chi phí để một nhân viên thực hiện một công việc.
* Đầu ra là một phân công công việc tối ưu, tức là danh sách các nhân viên được phân công cho từng công việc với tổng chi phí thấp nhất.

**Bước 2: Đặc tả yêu cầu**

1. **Các yêu cầu:**

* Tìm hiểu bài toán phân công công việc: Nắm vững định nghĩa, lý thuyết và các thuật toán phổ biến để giải bài toán này.
* Cài đặt thuật toán giải bài toán phân công công việc: Chọn và cài đặt một thuật toán cụ thể (vét cạn và nhánh cận).

1. **Yêu cầu chi tiết**

* Đầu vào: Một ma trận chi phí C với kích thước n x n, trong đó n là số công việc và số nhân viên.
* Đầu ra: Một danh sách phân công công việc tối ưu cùng với tổng chi phí.

1. **Yêu cầu về thuật toán**

* Hiệu quả: Chọn thuật toán phù hợp với kích thước ma trận.
* Độ phức tạp: Đánh giá độ phức tạp tính toán của từng thuật toán.

1. **Yêu cầu chức năng**

* Nhập ma trận chi phí: Chương trình phải có khả năng nhận đầu vào là ma trận chi phí.
* Tìm phân công công việc tối ưu: Chương trình cần có thuật toán tìm phân công công việc với chi phí thấp nhất.
* Kiểm tra tính khả thi: Trong quá trình tìm kiếm, chương trình cần kiểm tra xem phân công hiện tại có hợp lệ không.
* Hiển thị kết quả: Chương trình phải trả về phân công tối ưu và tổng chi phí.

1. **Yêu cầu phi chức năng**

* Hiệu suất: Chương trình cần xử lý các ma trận nhỏ đến vừa trong thời gian hợp lý.
* Tính chính xác: Đảm bảo độ chính xác trong việc tìm phân công công việc tối ưu.
* Thời gian phản hồi: Xử lý nhanh chóng với các ma trận nhỏ.

1. **Các phương pháp giải bài toán**

* Vét cạn
* Nhánh cận
* Tham lam (nếu áp dụng cho bài toán mở rộng)

**Bước 3: Thiết kế**

**MỤC LỤC  
DANH MỤC HÌNH ẢNH  
MỞ ĐẦU**

1. Lý do chọn đề tài
2. Mục tiêu tìm hiểu
3. Kết quả mong muốn
4. Tổng quan nội dung tìm hiểu

# TỔNG QUAN VỀ BÀI TOÁN PHÂN CÔNG CÔNG VIỆC **1.1. Định nghĩa về bài toán phân công công việc 1.2. Tầm quan trọng và ứng dụng của bài toán 1.3. Độ phức tạp và đặc điểm của bài toán 1.4. Ưu và nhược điểm của bài toán phân công công việc** **1.4.1. Ưu điểm     1.4.2. Nhược điểm**

# CÁC PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN **2.1. Thuật toán vét cạn (Brute Force) 2.2. Thuật toán nhánh cận (Branch and Bound) 2.3. Đánh giá hiệu quả các phương pháp**

# CÀI ĐẶT VÀ TRIỂN KHAI CÁC THUẬT TOÁN

## Mô tả yêu cầu kỹ thuật

### Các công cụ, ngôn ngữ lập trình và thư viện sử dụng để triển khai.

## Cài đặt thuật toán Vét cạn

### Mô tả chi tiết cách triển khai thuật toán Vét cạn.

### Kết quả thu được từ thuật toán.

## Cài đặt thuật toán Nhánh cận

### Mô tả chi tiết cách triển khai thuật toán Nhánh cận.

### Kết quả thu được.

# KẾT QUẢ VÀ ĐÁNH GIÁ

## So sánh kết quả giữa hai thuật toán

### Tổng chi phí và cách phân công.

### Thời gian thực thi với các kích thước ma trận khác nhau.

## Đánh giá hiệu quả thuật toán

### Phân tích độ phức tạp thực tế và các bài toán ứng dụng thực tế.

# KẾT LUẬN

1. Nội dung đã thực hiện
2. Hướng phát triển

**TÀI LIỆU THAM KHẢO**