



ALGORITMIA E ESTRUTURA DE DADOS

---

# Travel Salesman Problem

---

*Discente(s):*

Pedro ROLDAN

Leandro MOREIRA

*Docente:*

Doutor Faroq ALTAM

27 de Junho de 2017

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Requisitos Minimos . . . . .	2
1.1.1	RGB Decoder . . . . .	4
1.1.2	BCD - Binary Coded Decimal . . . . .	5
1.1.3	ROM Decoder . . . . .	7
<b>2</b>	<b>Enquadramento</b>	<b>8</b>
2.1	Motivação . . . . .	8
2.2	Objectivos . . . . .	8
<b>3</b>	<b>Conclusões</b>	<b>8</b>
<b>4</b>	<b>Bibliografia</b>	<b>9</b>
<b>5</b>	<b>Anexos</b>	<b>10</b>

# 1 Introdução

O travel salesman problem (TSP) é um problema bastante comum largamente encontrado em diversas aplicações tais como: empresas de transporte (e.g. UPS), escalas de tripulação de companhias aéreas, etc.

Em princípio, um vendedor necessita de efetuar uma viagem por diversas cidades, onde inicia a viagem numa determinada cidade (Casa), visita todas as cidades para vender os seus produtos, e retorna a casa.

## 1.1 Requisitos Minimos

O TSP pode ser representado por uma lista de nós, sendo o objectivo descobrir uma serie de caminhos (Edges) entre cada um dos nós.

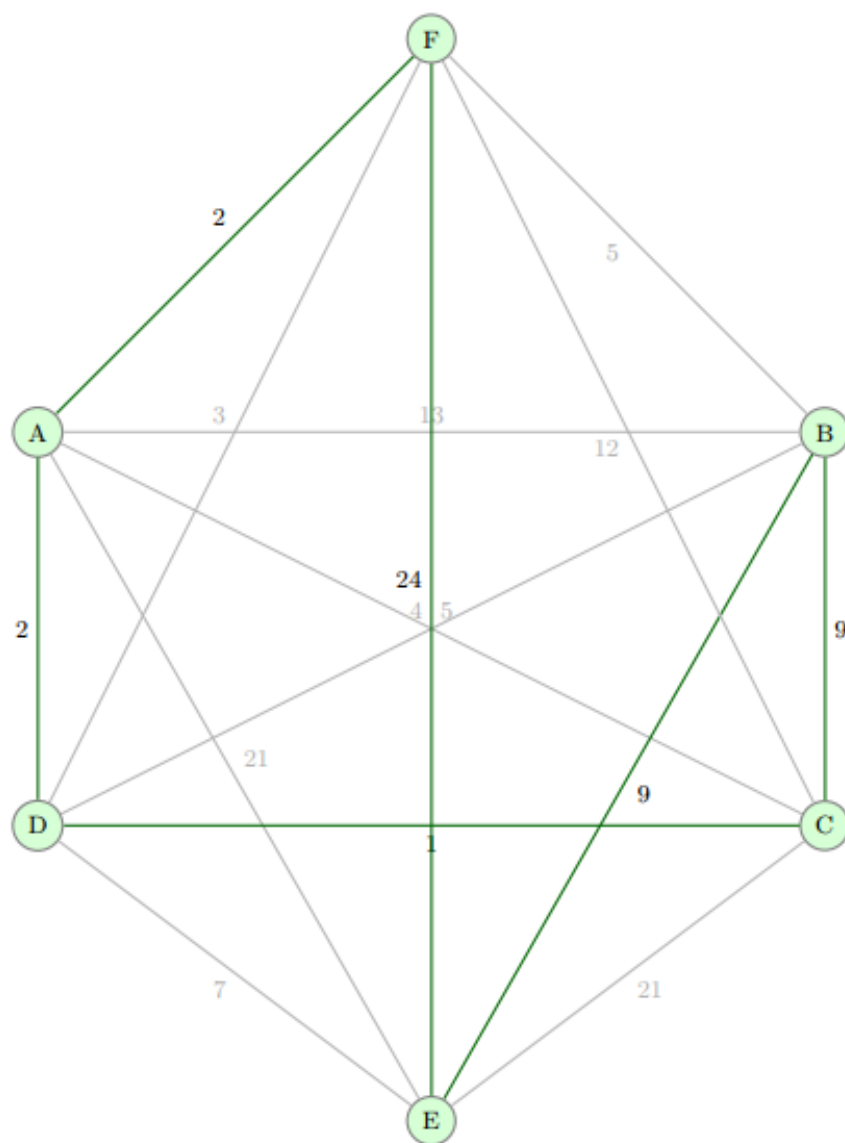
Sendo que:

- Cada nó (Cidade) pode ser visitado apenas 1 vez.
- Os caminhos formam uma Tour.
- O custo da Tour deve ser o minimo possivel.

A tour TSP é um grafico direcionado, onde cada nó representa uma cidade, e cada edge representa um caminho entre 2 cidades.

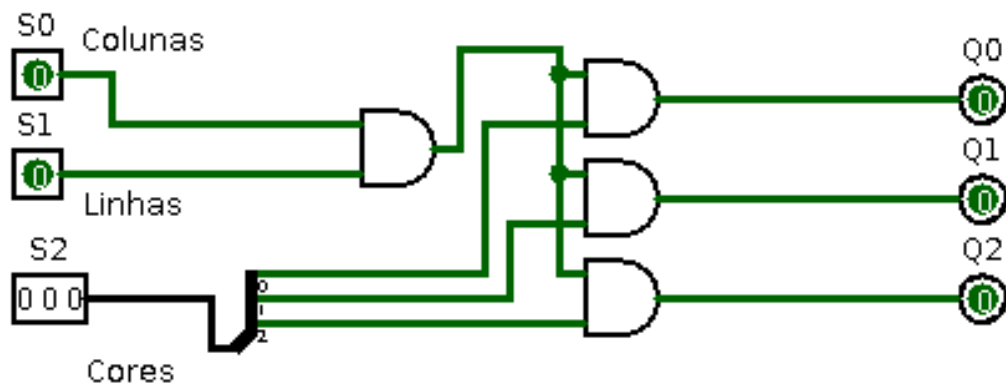
Cada edge têm um peso, que é no seu caso mais simples a distancia euclidiana entre os seus nós.

Este peso pode ser composto por diversos fatores, no entanto neste projeto apenas se considera a distancia entre nós.



### 1.1.1 RGB Decoder

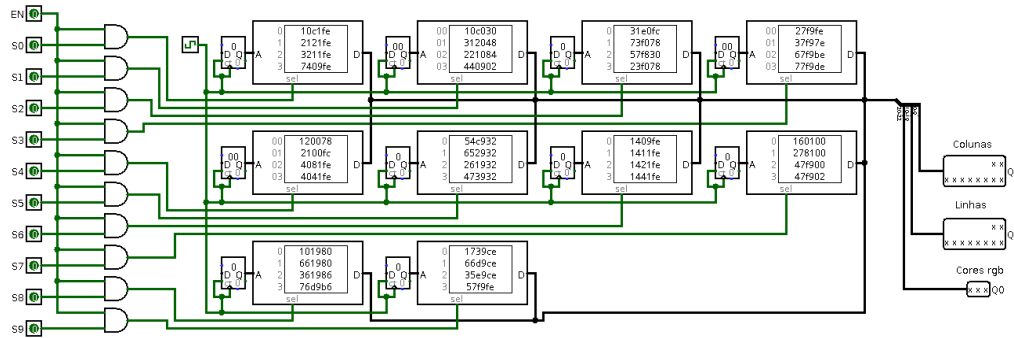
A função deste circuito é a de ao receber a indicação de posição através do grupo binário de S0 e S1, efetuar a correspondência de cor para a saída do led correspondente, através da separação do grupo de 3 bits.



**Figura 1:** RGB Decoder

S0-S3: Conjunto binário de 4 bits correspondendo ao seu valor decimal

A ROM (read-only memory), é um tipo de memória que permite apenas a leitura, ou seja, as suas informações são gravadas uma única vez e após isso não podem ser alteradas ou apagadas, somente acedidas. São memórias cujo conteúdo é gravado permanentemente.



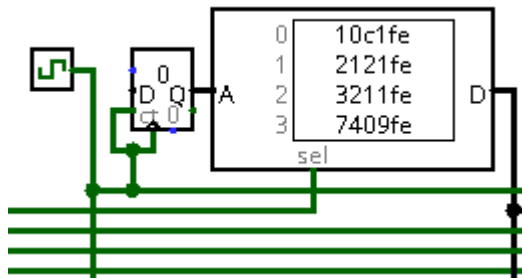
**Figura 2:** Circuito de controle de ROMS

EN: Bit de controlo de enable/disable

S0-S9: Bit de entrada para seleção de ROM

Q0: Conjunto de 3 bits para seleção de cor RGB

Cada ROM é controlada por um contador exclusivo e um clock partilhado.



**Figura 3:** ROM Individual

Cada ROM tem uma estrutura dimensionada para cada conjunto de sequencias ..

A ROM é controlada por um contador associado a um clock, que permite que, em cada ciclo o contador avance para uma nova posição (index) da ROM e efetue a leitura da sequencia de bits armazenada, que por sua vez define o endereço do led na matriz, assim como a sua cor e estado.

### 1.1.2 BCD - Binary Coded Decimal

O código BCD foi criado para codificar os números decimais de 0 a 9, com 4 bits para cada dígito, ou seja, o BCD é a conversão dos decimais em

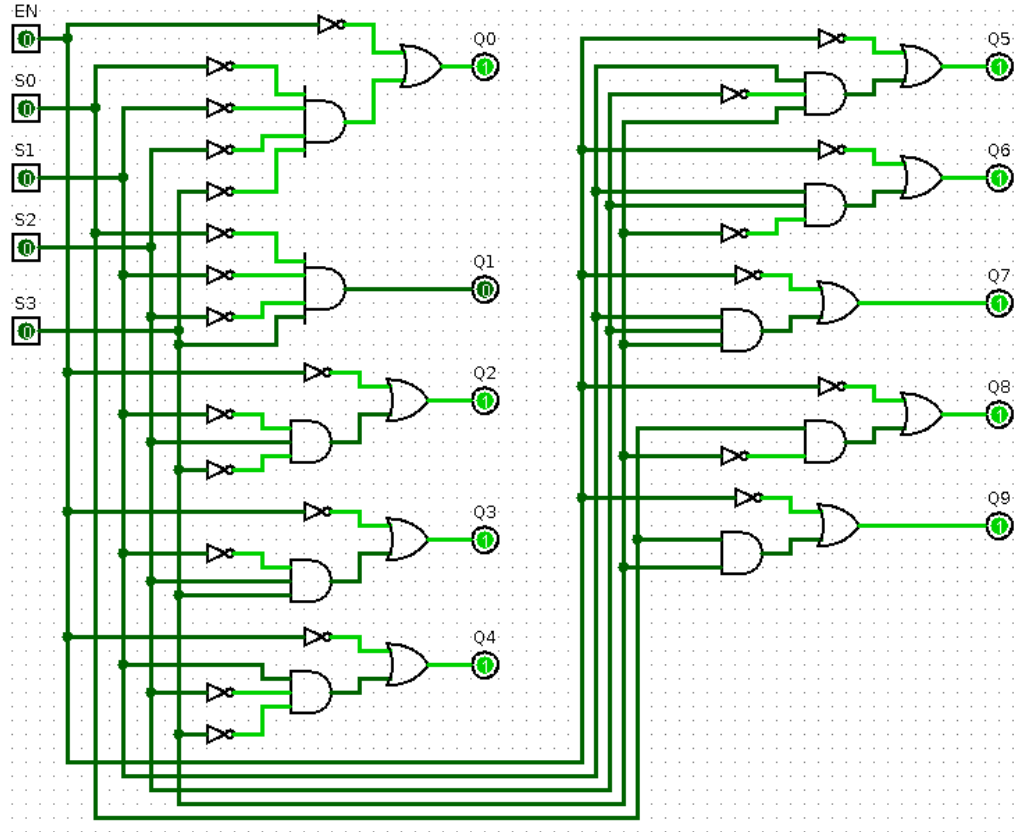
um número binário de 4 bits e representa-se da seguinte forma:

**Tabela 1:** Tabela BCD

Digito Decimal	Codigo BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Desta forma foi realizada a codificação segundo cada palavra do código BCD, que corresponde ao dígito decimal correspondente a cada ROM utilizada.

### 1.1.3 ROM Decoder



**Figura 4:** Circuito ROM Decoder

EN: Bit de controlo de enable/disable

S0-S3: Conjunto de 4 bits de entra que determina a escolha da ROM a ser utilizada

Q0-Q9: Bit de controlo de saída com escolha de ROM

**Tabela 2:** Tabela de Seleção de ROM

	S3	S2	S1	S0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	1	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	1	0	0	0	0	0
5	0	1	0	1	0	0	0	0	0	1	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0	1	0	0	0
7	0	1	1	1	0	0	0	0	0	0	0	1	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	1	0
9	1	0	0	1	0	0	0	0	0	0	0	0	0	1



## 2 Enquadramento

O trabalho descrito neste relatório foi realizado recorrendo à linguagem de programação C, assim como os recursos disponibilizados na unidade curricular.

### 2.1 Motivação

A principal motivação para a realização deste trabalho, resulta da importância em criar e otimizar um sistema digital, assim como demonstrar os conhecimentos alcançados na disciplina de sistemas digitais.

### 2.2 Objectivos

Pretende-se através deste trabalho, criar um sistema de luzes como Em ultima analise o sistema digital é um sistema electrónico onde os níveis de tensão eléctrica são mapeados como “0” e “1”.

Na saída do circuito encontram-se ligados LEDs que estarão acesos ou apagados com “1” ou “0”, respetivamente.

## 3 Conclusões

A matriz de leds desenvolvida, para além de permitir os requisitos pedidos no enunciado do trabalho prático, permite também o uso de animações com leds mais complexas, mais padrões disponíveis e sendo modular torna-se mais escalável, entre outras funcionalidades.

De frisar que devido à liberdade proporcionada para a construção dos circuitos, quer na sua forma de desenvolvimento quer na implementação permitiu desta forma aguçar a curiosidade para o uso de diversos componentes do simulador Logisim.

Foi sem duvida um desafio interessante, mas que por limitação de tempo, deixa ainda uma larga margem para melhoramentos.

## 4 Bibliografia

### Referências

- [1] <https://en.wikipedia.org/wiki/2-opt>.
- [2] [https://en.wikipedia.org/wiki/travelling salesman problem](https://en.wikipedia.org/wiki/travelling_salesman_problem).
- [3] Doutor Faroq AlTam. *Aulas Teórico-Práticas Algoritmia e Estruturas de Dados 2º ano, 2º semestre da Licenciatura em Engenharia Informática do Instituto Superior Manuel Teixeira Gomes*. ISMAT, 2016-2017.

## 5 Anexos

Ficheiro "relatorio.pdf" e "ledmatrix.circ" compactado num ficheiro "trabalho.zip".

Não existem quaisquer códigos ou listagens adicionais.