Trabalho Prático

Matemática Discreta

Autores:
Bruno Dylan Pinto Ferreira 8200586
Gonçalo André Fontes Oliveira 8200595
Jorge Miguel Fernandes Correia 8200592
Nuno de Figueiredo Brito e Castro 8200591

Grupo: 17

Licenciatura em Engenharia Informática
2020/2021

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Índice

Introdução

No âmbito da unidade curricular de matemática discreta foi-nos proposta a resolução de 3 questões relacionadas com a matéria lecionada ao longo do 2º semestre como:

- Indução Matemática ou aplicação do algoritmo EGV(Expand, Guess, Verify): O Método de Indução Matemática é um método que tem por base o Princípio de Indução Finita que é bastante utilizado para comprovar se determinadas propriedades são verdadeiras tendo em conta o conjunto dos números naturais.

- Teoria dos grafos é um ramo da matemática que estuda as relações entre os objetos de um determinado conjunto. Para tal são empregadas estruturas chamadas de grafos, G(V,E), onde V é um conjunto não vazio de objetos denominados vértices e E é um subconjunto de pares não ordenados de V denominados de arestas.

- Em criptografia, **encriptação** é o processo de transformar informação usando um algoritmo, chamado de cifra, de modo a impossibilitar a sua leitura. Por isso, ao processo de tornar informação encriptada novamente legível dá-se o nome de **desencriptação**.

Este relatório está estruturado em 3 partes onde damos solução aos diversos exercícios propostos no enunciado explicando toda a resolução realizada até ao resultado final.

Começamos por resolver um exercício que aborda a aplicação do método EGV onde começamos por apresentar a nossa função e os respetivos passos dados até à sua resolução final. Em cada passo além dos cálculos pode ser visualizada uma breve explicação do que foi realizado o que torna mais fácil a perceção do que foi feito no exercício.

Seguidamente resolvemos um exercício referente à matéria de Teoria dos Grafos. Neste exercício aplicamos diversos conhecimentos desta matéria que fomos obtendo ao longo do semestre. O exercício envolve monumentos e alojamentos da Rota do Românico e consiste em calcularmos caminhos e/ou circuitos obedecendo a determinadas condições e utilizando diferentes algoritmos.

Por fim, resolvemos um exercício que diz respeito à matéria de Criptografia que está subdividido em duas partes: Encriptação e Desencriptação. Neste exercício necessitamos de completar uma frase que depois é encriptada utilizando algoritmos. Após a encriptação dessa mensagem é necessário desencriptá-la para que esta volte a ser legível.

Nota: Ao longo da realização deste trabalho prático tivemos o cuidado de apresentar o uso de um algoritmo através de cálculos realizados por nós e através do uso de um algoritmo feito em Python, desenvolvido também por nós. O uso da linguagem Python para desenvolver um algoritmo que fosse capaz de resolver o exercício foi uma forma muito mais fácil e eficaz sendo que nos poupa bastante trabalho e tempo em cálculos e, além disso a taxa de erro na resolução dos cálculos é muito menor.

Com a realização deste trabalho vamos aprofundar os conceitos lecionados durante o $2^{\underline{o}}$ semestre na unidade curricular de Matemática Discreta. Estes conceitos serão importantes para o nosso futuro dado que podem ser vir a utilizados no desenvolvimento de futuros projetos.

Resolução dos exercícios

1. Exercício EGV

Nesta questão é proposto fazer a aplicação do algoritmo EGV(Expand, Guess, Verify = Desenvolver, Estimar, Verificar) que consiste em encontrar a sua fórmula fechada.

$$\begin{cases} f(1) = 2 \\ f(n) = 2 \times f(n-1) + 2^n \text{ para } n \geqslant 2 \end{cases}$$

Expand - Este passo consiste em utilizar a fórmula de recorrência da função apresentada e expandi-la da seguinte forma:
Começar por $f(n)$, que é função de $f(n-1)$:
Calcular $f(n-1)$ e substituir em $f(n)$, obtendo-se assim $f(n)$ em função de $f(n-2)$.
Calcular $f(n-2)$ e substituir em $f(n)$, obtendo-se assim $f(n)$ em função de $f(n-3)$.
Calcular $f(n-3)$ e substituir em $f(n)$, obtendo-se assim $f(n)$ em função de $f(n-4)$.

Tendo isto em conta obtemos que:
$f(n) = 2 \times f(n-1) + 2^n$
$= 2 \times (2 \times f(n-2) + 2^{n-1}) + 2^n = 2^2 \times f(n-2) + 2^n + 2^n = 2^2 \times f(n-2) + 2 \times 2^n$
$= 2^2(2 \times f(n-3) + 2^{n-2}) + 2 \times 2^n = 2^3 \times f(n-3) + 3 \times 2^n$
$= 2^3(2 \times f(n-4) + 2^{n-3}) + 3 \times 2^n = 2^4 \times f(n-4) + 4 \times 2^n$

Guess - Neste passo utilizou-se a fórmula final obtida através do passo Expand para mostrar qual é a nossa definição, onde k representa o nosso argumento. Em seguida consideramos que $n - k = 1$ logo $k = n - 1$
Então, podemos conjeturar que:

$f(n) = 2^k \times f(n-k) + k \times 2^n$

Se $n - k = 1 (=) k = n - 1$, temos

$f(n) = 2^{n-1} \times f(n - (n-1)) + (n-1) \times 2^n$
$= 2^{n-1} \times f(1) + (n-1) \times 2^n$
$= 2^{n-1} \times 2 + n \times 2^n - 2^n$
$= 2^n + n \times 2^n - 2^n$
$= n \times 2^n$

Verify - Por fim podemos verificar que a nossa fórmula fechada está correta através de uma demonstração indutiva, que se divide em dois passos:

Passo Base - Neste passo pretendemos mostrar que a fórmula se verifica para n = 1, sendo 1 um valor inicial. Por isso, fazendo os respetivos cálculos podemos observar que os valores se confirmam:

$$f(1) = 1 \times 2^1 = 2 \textbf{ Verdadeiro}$$

Passo indutivo (Hipótese) - Neste passo temos de mostrar que se a fórmula se confirma para n = k, então a mesma fórmula confirma-se para n = k + 1, ou seja, provando assim que o processo usado na passagem de um valor para o valor seguinte é válido. Supondo que a expressão n = k é verdadeira, substitui-se os n's por k e assim apresentamos a nossa Hipótese:

$$f(k) = k \times 2^k$$

Passo indutivo (Tese) - De seguida é necessário provar a que a nossa tese é válida, começando por igualar $f(k+1)$ à fórmula de recorrência com as respetivas substituições de n por k + 1.

$$\begin{aligned} f(k+1) &= 2 \times f(k+1-1) + 2^{k+1} = 2 \times f(k) + 2^{k+1}, \text{ por definição de f} \\ &= 2 \times k \times 2^k \times 2^{k+1}, \text{ pela hipótese de indução} \\ &= k \times 2^{k+1} \times 2^{k+1} = (k+1) \times 2^{k+1}, \text{ c.q.m} \end{aligned}$$

2. Exercício Teoria dos grafos

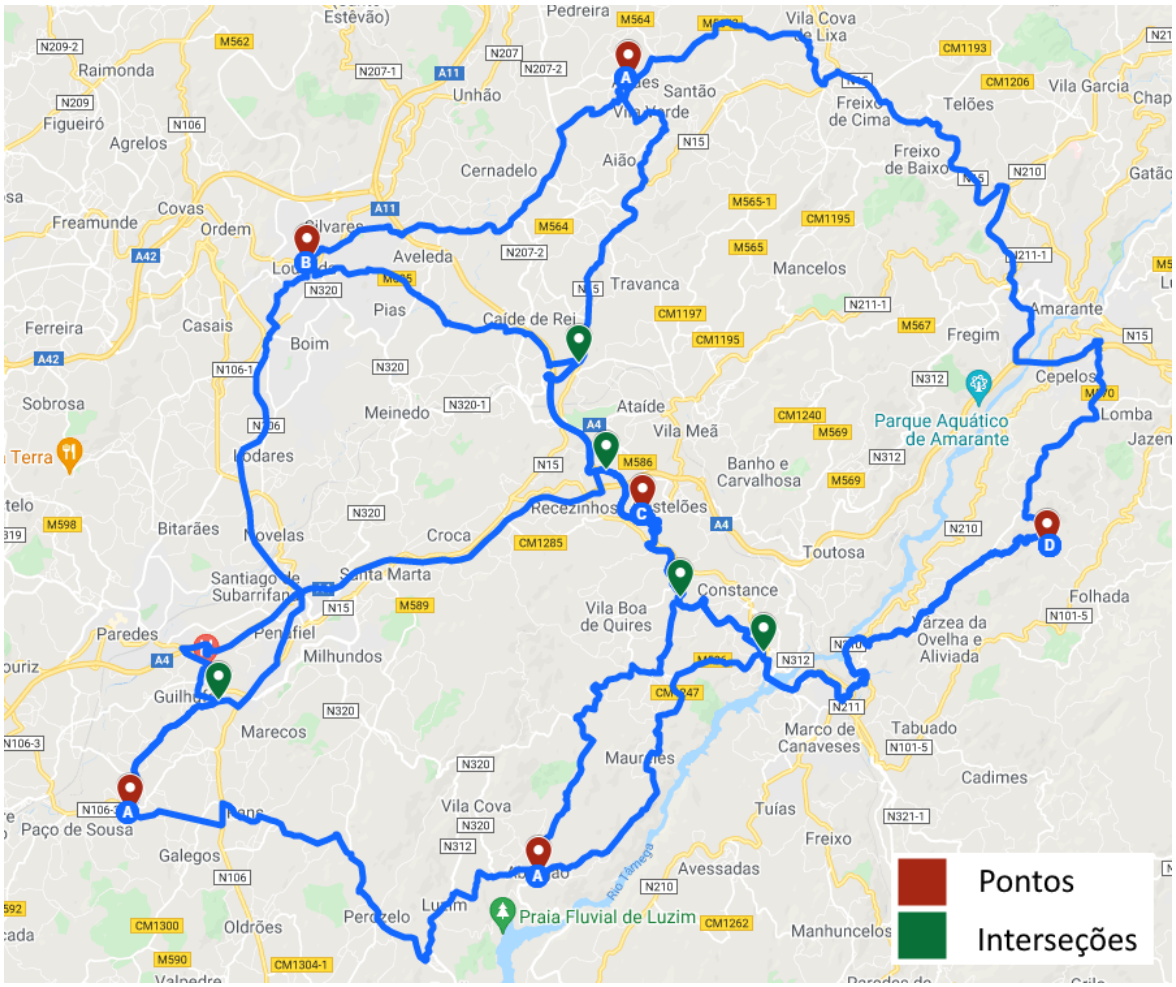
Neste exercício é proposto que criemos um grafo com 6 vértices em que 5 representam monumentos da Rota do Românico e o 6º representa o alojamento para pernoitar que deve estar localizado dentro da região da Rota do Românico.Decidimos criar um percurso no qual os nossos visitantes desfrutem dos seguintes locais Ponto do arco, Memorial da Ermida, Igreja de Santa Maria de Airães, Centro de Interpretação do românico, Igreja de São Pedro de Abragão e tivemos em conta número de kms percorridos, consumo de combustível (gasóleo com o custo de 1,283€por litro) e portagens (caso existam).

Para a elaboração do exercício foram usadas abreviaturas para definir os pontos e também foram levados em consideração 3 critérios:

1. Abreviaturas:
    - Casas do Telhado (Penafiel) - CT;
    - Ponto do arco (Marco Canaveses) - PA;
    - Memorial da Ermida (Penafiel) - ME;
    - Igreja de Santa Maria de Airães (Felgueiras) - ISMA;
    - Centro de Interpretação do românico (Lousada) - CI;
    - Igreja de São Pedro de Abragão (Penafiel) - ISPA;
    - Interseções- IN.
2. Critérios:
    - Número de kms percorridos;
    - Consumo de combustível e portagens (caso existam);
    - Tempo despendido na viagem.

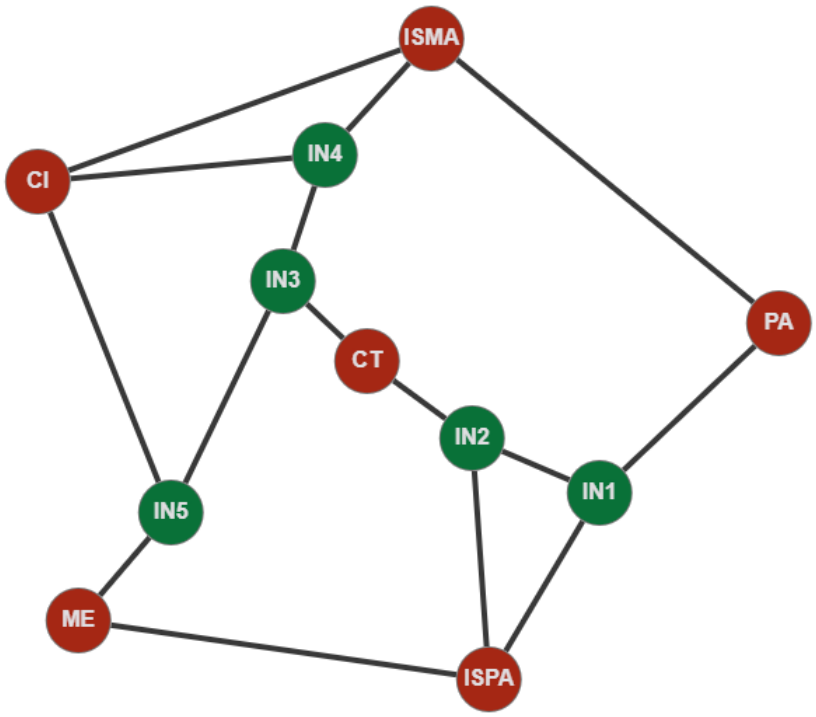
Representação real do Grafo

É possível aceder a esta representação pelo link seguinte: https://www.google.com/maps/d/u/0/edit?mid=1gjb0kchVgrH6ip_8DapSzaq5lc3uEJs0&usp=sharing



Representação do Grafo

Para resolver o problema acima foi então criado um novo vértice (Inter) como podemos ver abaixo



- V(G) = {ISMA, CT, PA, ISPA, ME, CI, IN1, IN2, IN3, IN5, IN4}
- E(G) = {(ISMA, CI), (ISMA, IN4), (ISMA, PA), (CI, IN4), (CI, IN5), (IN4, IN3), (PA, IN1),(IN5, ME), (IN5, IN3), (IN3, CT), (IN1, IN2), (IN1, ISPA), (ME, ISPA), (CT, IN2), (IN2, ISPA)}
- Ordem do Grafo - 11 vértices
- Dimensão do Grafo - 15 arestas

Matriz de adjacência

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{array}{l} \text{Grau(ISMA)} = 3 \\ \text{Grau(CT)} = 2 \\ \text{Grau(PA)} = 2 \\ \text{Grau(ISPA)} = 3 \\ \text{Grau(ME)} = 2 \\ \text{Grau(CI)} = 3 \\ \text{Grau(IN1)} = 3 \\ \text{Grau(IN2)} = 3 \\ \text{Grau(IN3)} = 3 \\ \text{Grau(IN5)} = 3 \\ \text{Grau(IN4)} = 3 \end{array}$$

Ordem dos vértices: ISMA, CT, PA, ISPA, ME, CI, IN1, IN2, IN3, IN5, IN4

Classificação do grafo

1. Grafo Simples.

   - O grafo é simples porque cada aresta liga dois vértices diferentes, nenhuma aresta liga um vértice a si mesmo e não existem duas arestas diferentes que ligam os mesmos vértices.

2. Grafo não orientado

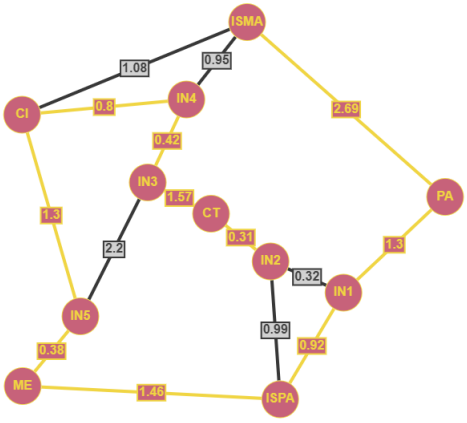
   - O grafo é não orientado porque as arestas não possuem direções, por isso a matriz de adjacências é simétrica

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Ordem dos vértices: ISMA, CT, PA, ISPA, ME, CI, IN1, IN2, IN3, IN5, IN4

3. Grafo Semi-Hamiltoniano

- O grafo semi-Hamiltoniano é um grafo que admite um caminho de Hamilton.
  - Um caminho de Hamilton é um caminho num grafo que passa em cada vértice exatamente uma vez.
- **Exemplo de um caminho de hamilton do grafo**: (ISMA, PA, IN1, ISPA, ME, IN5, CI, IN4, IN3, CT, IN2)



4. Grafo conexo e fortemente conexo

- Um grafo não orientado diz-se conexo se, para cada par de vértices, existe um caminho que os une. Caso contrário, o grafo diz-se desconexo.
- Pode-se dizer também que um grafo não orientado é conexo (fortemente conexo) se o fecho transitivo da sua matriz de adjacências não tiver entradas nulas.
- O fecho transitivo de uma matriz de adjacências M, de ordem n, de um grafo G é a matriz F definida por:

$$F = M + M^2 + M^3 + M^4 + M^5 + M^6 + M^7 + M^8 + M^9 + M^{10} + M^{11}$$

$$= \begin{bmatrix}
16647 & 9197 & 10009 & 10722 & 9197 & 18353 & 10695 & 10722 & 15430 & 15430 & 18353 \\
9197 & 5459 & 6352 & 7849 & 7011 & 10824 & 8381 & 9016 & 10173 & 7649 & 8976 \\
10009 & 6352 & 6398 & 8287 & 6352 & 10472 & 8615 & 8287 & 9291 & 9291 & 10472 \\
10722 & 7849 & 8287 & 12016 & 9016 & 11842 & 12532 & 12887 & 11521 & 9673 & 10485 \\
9197 & 7011 & 6352 & 9016 & 5459 & 8976 & 8381 & 7849 & 7649 & 10173 & 10824 \\
18353 & 10824 & 10472 & 11842 & 8976 & 18865 & 11419 & 10485 & 15337 & 18352 & 21093 \\
10695 & 8381 & 8615 & 12532 & 8381 & 11419 & 12277 & 12532 & 10685 & 10685 & 11419 \\
10722 & 9016 & 8287 & 12887 & 7849 & 10485 & 12532 & 12016 & 9673 & 11521 & 11842 \\
15430 & 10173 & 9291 & 11521 & 7649 & 15337 & 10685 & 9673 & 12411 & 16487 & 18352 \\
15430 & 7649 & 9291 & 9673 & 10173 & 18352 & 10685 & 11521 & 16487 & 12411 & 15337 \\
18353 & 8976 & 10472 & 10485 & 10824 & 21093 & 11419 & 11842 & 18352 & 15337 & 18865
\end{bmatrix}$$

- Podemos concluir assim que o grafo é conexo e fortemente conexo, pois se grafo não orientado é conexo logo é também fortemente conexo.

Nota: O grafo não é euleriano ou semi-euleriano porque não admite um circuito ou caminho de Euler, respetivamente. Além disso os graus não são todos pares nem existem apenas 2 graus ímpares.

Tabela dos critérios para o custo

Nesta tabela encontram-se todos os caminhos que ligam os diferentes vértices e, o seu respetivo custo para que fosse mais fácil identificar o mais "rentável". Para estes cálculos usamos a os dados fornecidos pela a aplicação Waze.

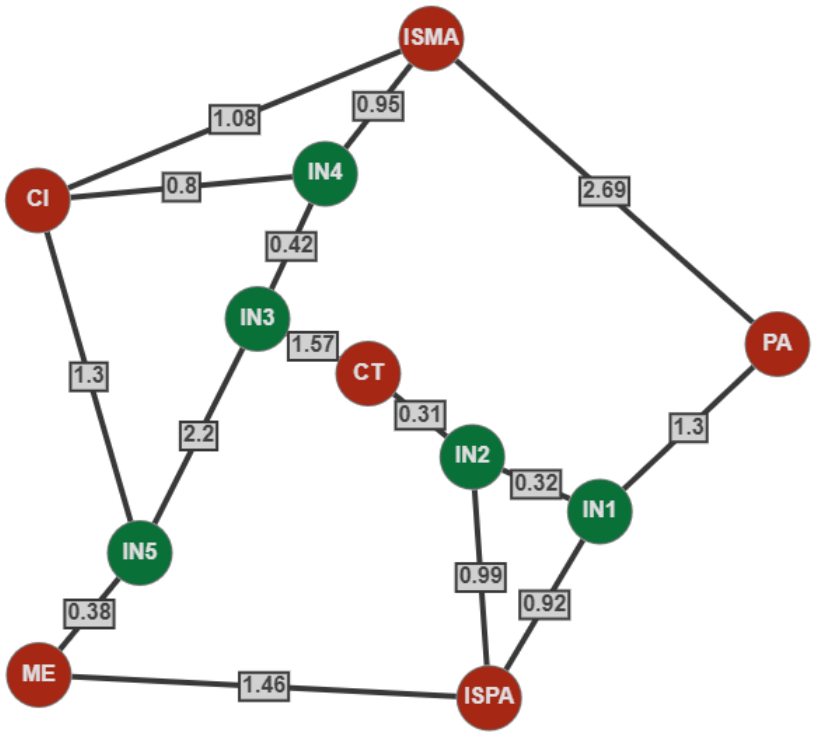
Para este exercício consideramos que um automóvel gasta 1 litro de gasóleo a cada 12 Km. Dado que o preço do gasóleo no momento se encontra a 1,283€ por litro, o preço do gasóleo por Km é $\frac{1,283}{12}$.

$$\text{Km} \times \frac{1,283}{12} + \text{portagens}$$

Vértice 1	**Vértice 2**	**km**	**Tempo viagem**	**Preço gasóleo gasta**	**Preço portagens**	**Total preço**
PA	ISMA	25,2 km	35 min	2,69€	Sem portagem	2,69€
PA	IN1	12,2 km	21 min	1,30€	Sem portagem	1,30€
ME	ISPA	13,7 km	22 min	1,46€	Sem portagem	1,46€
ME	IN5	3,6 km	6 min	0,38€	Sem portagem	0,38€
ISMA	CI	10,1 km	18 min	1,08€	Sem portagem	1,08€
ISMA	IN4	8,8 km	14 min	0,95€	Sem portagem	0,95€
CI	IN4	7,5 km	13 min	0,80€	Sem portagem	0,80€
CI	IN5	12,2 km	19 min	1,30€	Sem portagem	1,30€
ISPA	IN1	8,6 km	18 min	0,92€	Sem portagem	0,92€
ISPA	IN2	9,3 km	23 min	0,99€	Sem portagem	0,99€
CT	IN3	10 km	9 min	1,07€	0,50	1,57€
CT	IN2	2,9 km	5 min	0,31€	Sem portagem	0,31€
IN1	IN2	3 km	5 min	0,32€	Sem portagem	0,32€
IN3	IN4	3,9 km	3 min	0,42€	0,25	0,42€
IN3	IN5	20,6 km	16 min	2,20€	1,35	2,20€

Representação do Grafo com custos

Neste grafo foram adicionados os custos



Matriz de custos

$$\begin{bmatrix}
0 & \infty & 2,69 & \infty & \infty & 1,08 & \infty & \infty & \infty & \infty & 0,95 \\
\infty & 0 & \infty & \infty & \infty & \infty & \infty & 0,31 & 1,57 & \infty & \infty \\
2,69 & \infty & 0 & \infty & \infty & \infty & 1,3 & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & 0 & 1,46 & \infty & 0,92 & 0,99 & \infty & \infty & \infty \\
\infty & \infty & \infty & 1,46 & 0 & \infty & \infty & \infty & \infty & 0,38 & \infty \\
1,08 & \infty & \infty & \infty & \infty & 0 & \infty & \infty & \infty & 1,3 & 0,8 \\
\infty & \infty & 1,3 & 0,92 & \infty & \infty & 0 & 0,32 & \infty & \infty & \infty \\
\infty & 0,31 & \infty & 0,99 & \infty & \infty & 0,32 & 0 & \infty & \infty & \infty \\
\infty & 1,57 & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 2,2 & 0,42 \\
\infty & \infty & \infty & \infty & 0,38 & 1,3 & \infty & \infty & 2,2 & 0 & \infty \\
0,95 & \infty & \infty & \infty & \infty & 0,8 & \infty & \infty & 0,42 & \infty & 0
\end{bmatrix}$$

Ordem dos vértices: ISMA, CT, PA, ISPA, ME, CI, IN1, IN2, IN3, IN5, IN4

a) O percurso com menor custo com origem no alojamento e destino no monumento mais distante dos 5 escolhidos (não necessariamente passando por todos)

Vértice 1	Vértice 2	Caminho	Km
CT	ISMA	(CT,IN3,IN4,ISMA)	22,7 Km
CT	PA	(CT,IN2,IN1,PA)	18,1 Km
CT	ISPA	(CT,IN2,ISPA)	12,2 Km
CT	**ME**	**(CT,IN3,IN5,ME)**	**34.2 Km**
CT	CI	(CT,IN3,IN4,CI)	21,4 Km

Como é possível visualizar na tabela acima, o monumento com maior distância do local de alojamento é o Memorial da Ermida (ME) com 34.2 Km.

Após obtido o monumento mais distante do alojamento, é pedido para encontrar o percurso com menor custo com origem no alojamento e destino no monumento ME (Memorial da Ermida). Para isso utilizamos o algoritmo de Dijkstra:

O algoritmo de Dijkstra, concebido pelo cientista da computação holandês Edsger Dijkstra em 1956 e publicado em 1959, soluciona o problema do caminho mais curto num grafo orientado ou não orientado com arestas de peso não negativo, em tempo computacional onde V é o número de vértices e E é o número de arestas.

	Passo 1	Passo 2	Passo 3	Passo 4	Passo 5	Passo 6	Passo 7	Passo 8
CT	**(O, CT)**	∞	∞	∞	∞	∞	∞	∞
IN3	(1.57, CT)	(1.57, CT)	(1.57, CT)	(1.57, CT)	**(1.57, CT)**	∞	∞	∞
IN4	∞	∞	∞	∞	∞	(1.99, IN3)	**(1.99, IN3)**	∞
ISMA	∞	∞	∞	∞	∞	∞	(4.62, PA)	(2.94, IN4)
PA	∞	∞	∞	(1.93, IN1)	(1.93, IN1)	**(1.93, IN1)**	∞	∞
IN1	∞	(0.63, IN2)	**(0.63, IN2)**	∞	∞	∞	∞	∞
IN2	(0.31, CT)	**(0.31, CT)**	∞	∞	∞	∞	∞	∞
ISPA	∞	(1.3, IN2)	(1.3, IN2)	~~(1.55, IN1)~~ (1.3, IN2)	∞	∞	∞	∞
ME	∞	∞	∞	∞	(2.76, ISPA)	(2.76, ISPA)	(2.76, ISPA)	**(2.76, ISPA)**
IN5	∞	∞	∞	∞	∞	(3.77, IN3)	(3.77, IN3)	(3.77, IN3)
CI	∞	∞	∞	∞	∞	∞	∞	(2.79, IN4)

Caminho de menor custo de CT para ME :

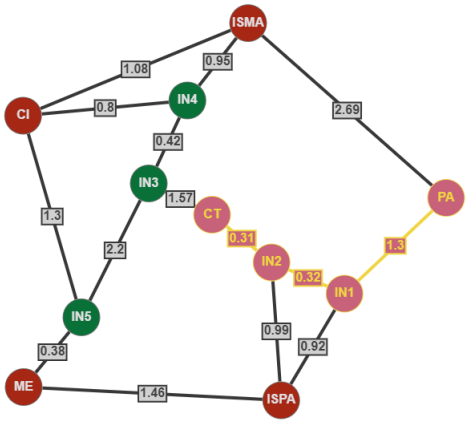
1. Fazer o caminho de ME para CT : ME, ISPA, IN2, CT
2. Colocar o caminho do ultimo vértice para o primeiro vértice: CT, IN2, ISPA, ME

b) O percurso com menor custo entre cada dois locais

Para tornar o processo do algoritmo de Dijkstra, mais fácil e rápido decidimos criar um programa em Python.

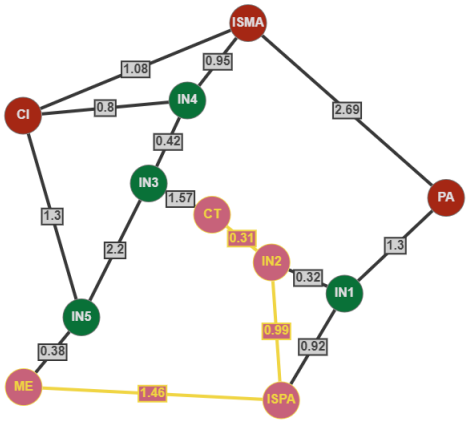
Este programa recebe o número de vértices, os nomes dos vértices, os seus pesos (como o nosso grafo é não orientado apenas é lida a diagonal inferior), por fim é pedido o vértice inicial e o vértice final. Para criar o programa utilizamos a seguinte biblioteca https://gitlab.com/AlanDeSmet/dijkstra-spf/-/tree/master que nos dá o caminho de Dijkstra num gráfico não orientado.

Casas do telhado para Ponto do arco:



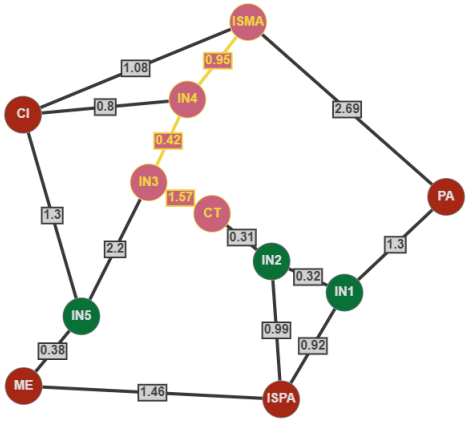
- Percurso: CT → IN2 → IN1 → PA
- Custo final: 1,93

Casas do telhado para Memorial da Ermida:



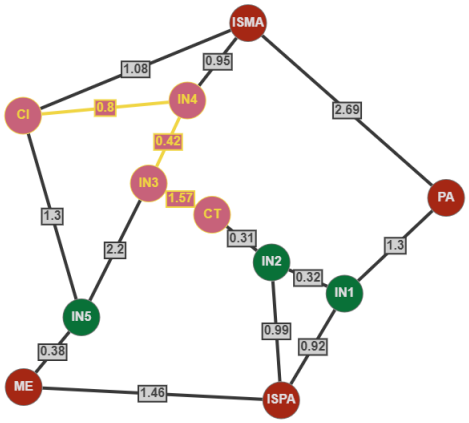
- Percurso: CT → IN2 → ISPA → ME
- Custo final: 2,76

Casas do telhado para Igreja de Santa Maria de Airães:



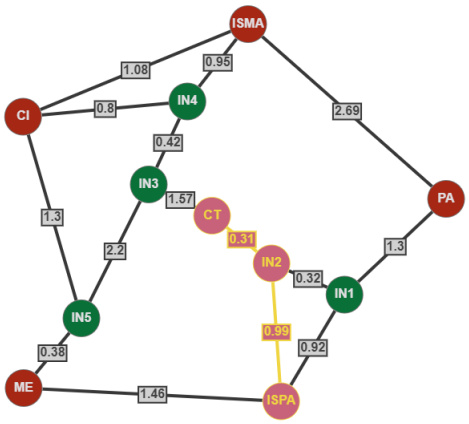
- Percurso: CT → IN3 → IN4 → ISMA
- Custo final: 2,94

Casas do telhado para Centro de Interpretação do românico:



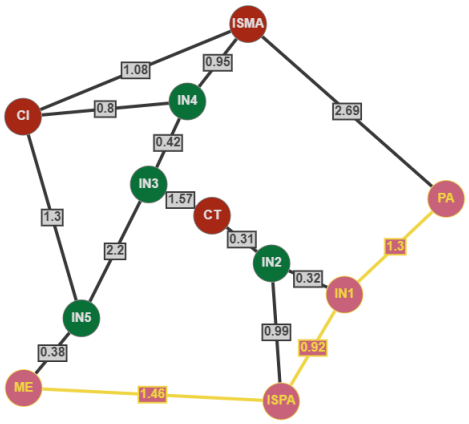
- Percurso: CT → IN3 → IN4 → CI
- Custo final: 2,79

Casas do telhado para Igreja de São Pedro de Abragão:



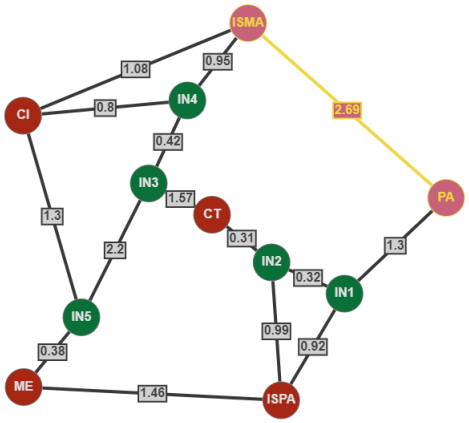
- Percurso: CT → IN2 → ISPA
- Custo final: 1,3

Ponto do arco para Memorial da Ermida:



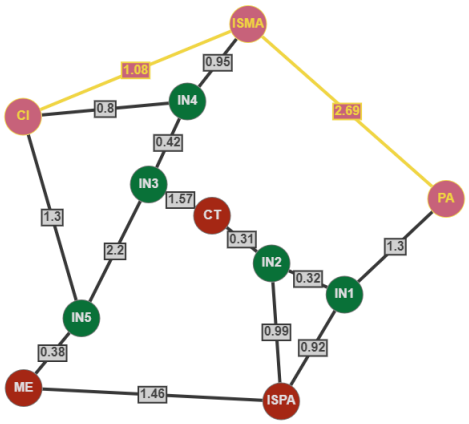
- Percurso: PA → IN1 → ISPA → ME
- Custo final:3,68

Ponto do arco para Igreja de Santa Maria de Airães:



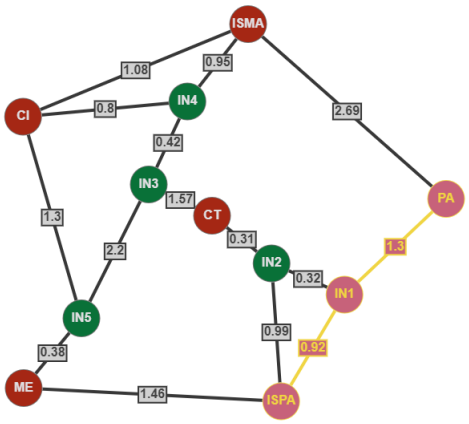
- Percurso: PA → ISMA
- Custo final: 2,69

Ponto do arco para Centro de Interpretação do românico:



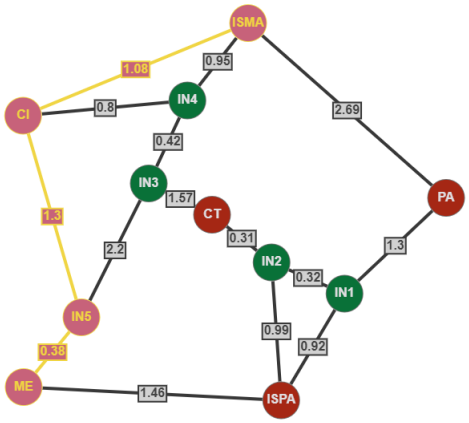
- Percurso: PA → ISMA → CI
- Custo final: 3,77

Ponto do arco para Igreja de São Pedro de Abragão:



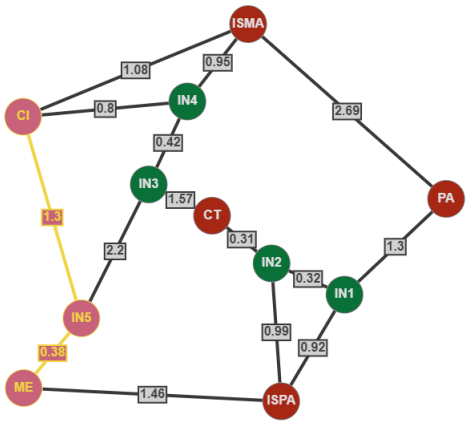
- Percurso: PA → IN1 → ISPA
- Custo final: 2.22

Memorial da Ermida para Igreja de Santa Maria de Airães:



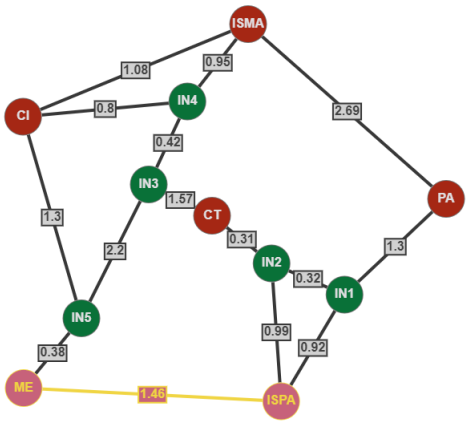
- Percurso: ME → IN5 → CI → ISMA
- Custo final: 2.76

Memorial da Ermida para Centro de Interpretação do românico:



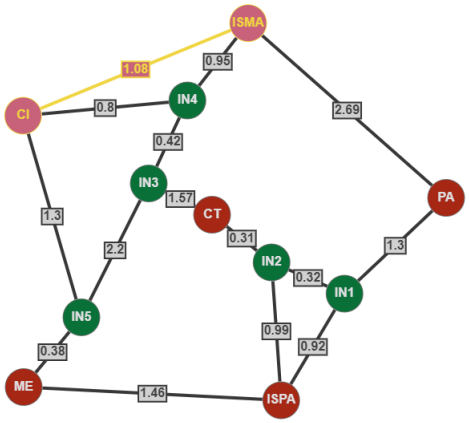
- Percurso: ME → IN5 → CI
- Custo final: 1,68

Memorial da Ermida para Igreja de São Pedro de Abragão:



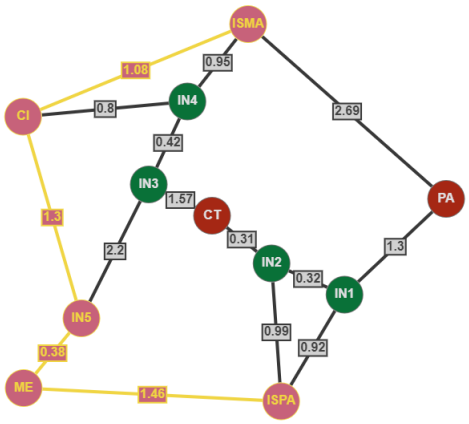
- Percurso: ME → ISPA
- Custo final: 1,46

Igreja de Santa Maria de Airães para Centro de Interpretação do românico:



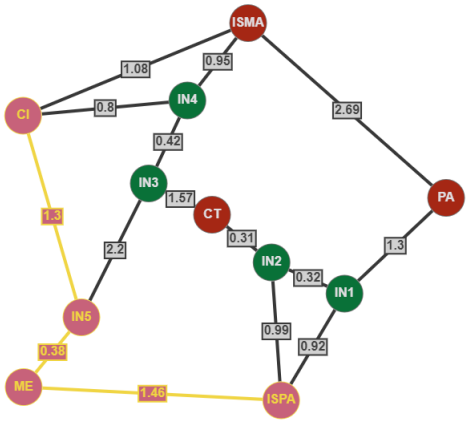
- Percurso: ISMA → CI
- Custo final: 1,08

Igreja de Santa Maria de Airães para Igreja de São Pedro de Abragão:



- Percurso: ISMA → CI → IN5 → ME → ISPA
- Custo final: 4,22

Centro de Interpretação do românico para Igreja de São Pedro de Abragão:



- Percurso: CI → IN5 → ME → ISPA
- Custo final: 3.14

Tabela com os custos entre os 5 Monumentos

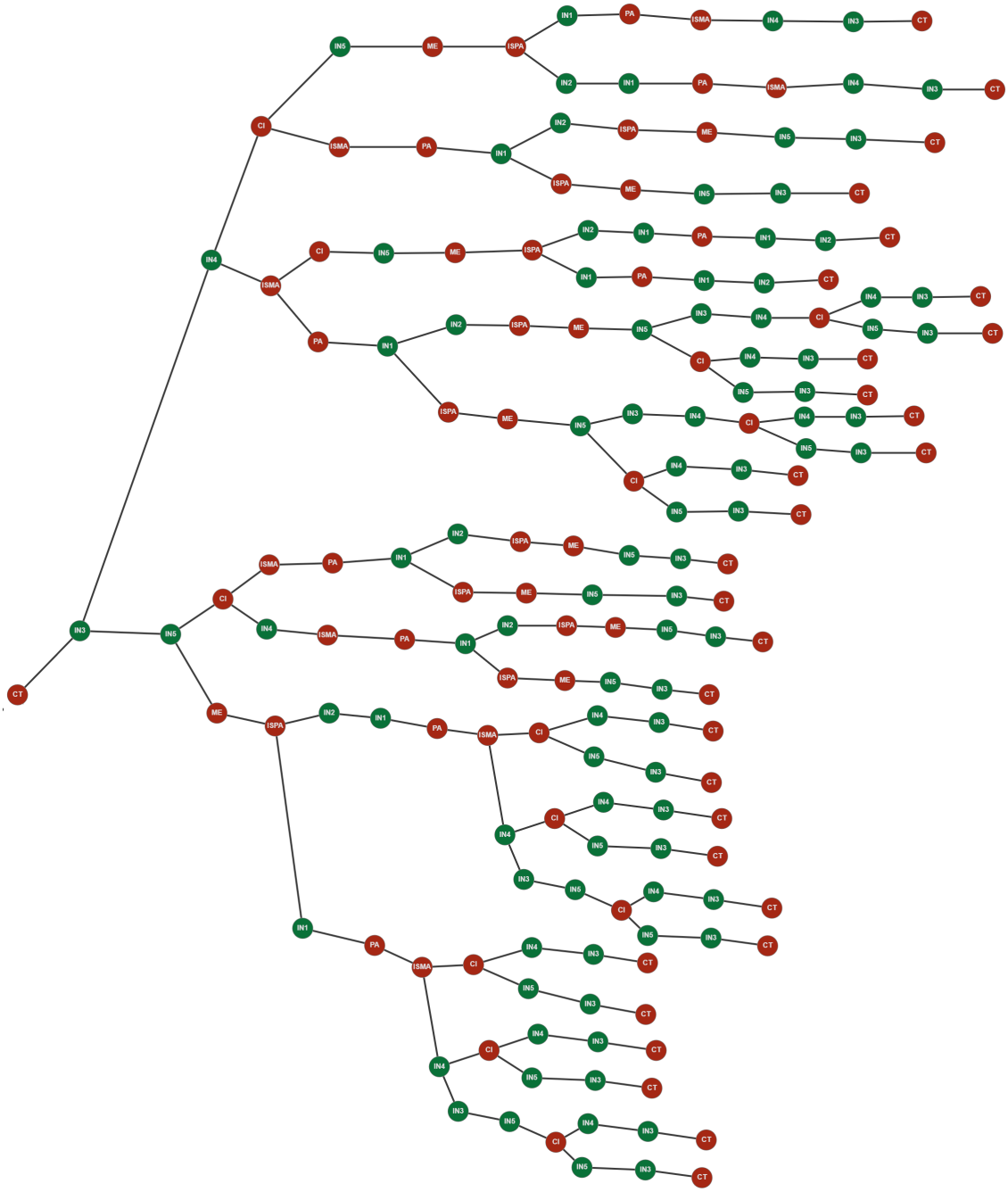
Monumentos	CT	ISPA	ME	CI	ISMA	PA
CT	0	1.3	2.76	2.79	2.94	1.93
ISPA	1.3	0	1.46	3.14	4.22	2.22
ME	2.76	1.46	0	1.68	2.76	3.68
CI	2.79	3.14	1.68	0	1.08	3.77
ISMA	2.94	4.22	2.76	1.08	0	2.69
PA	1.93	2.22	3.68	3.77	2.69	0

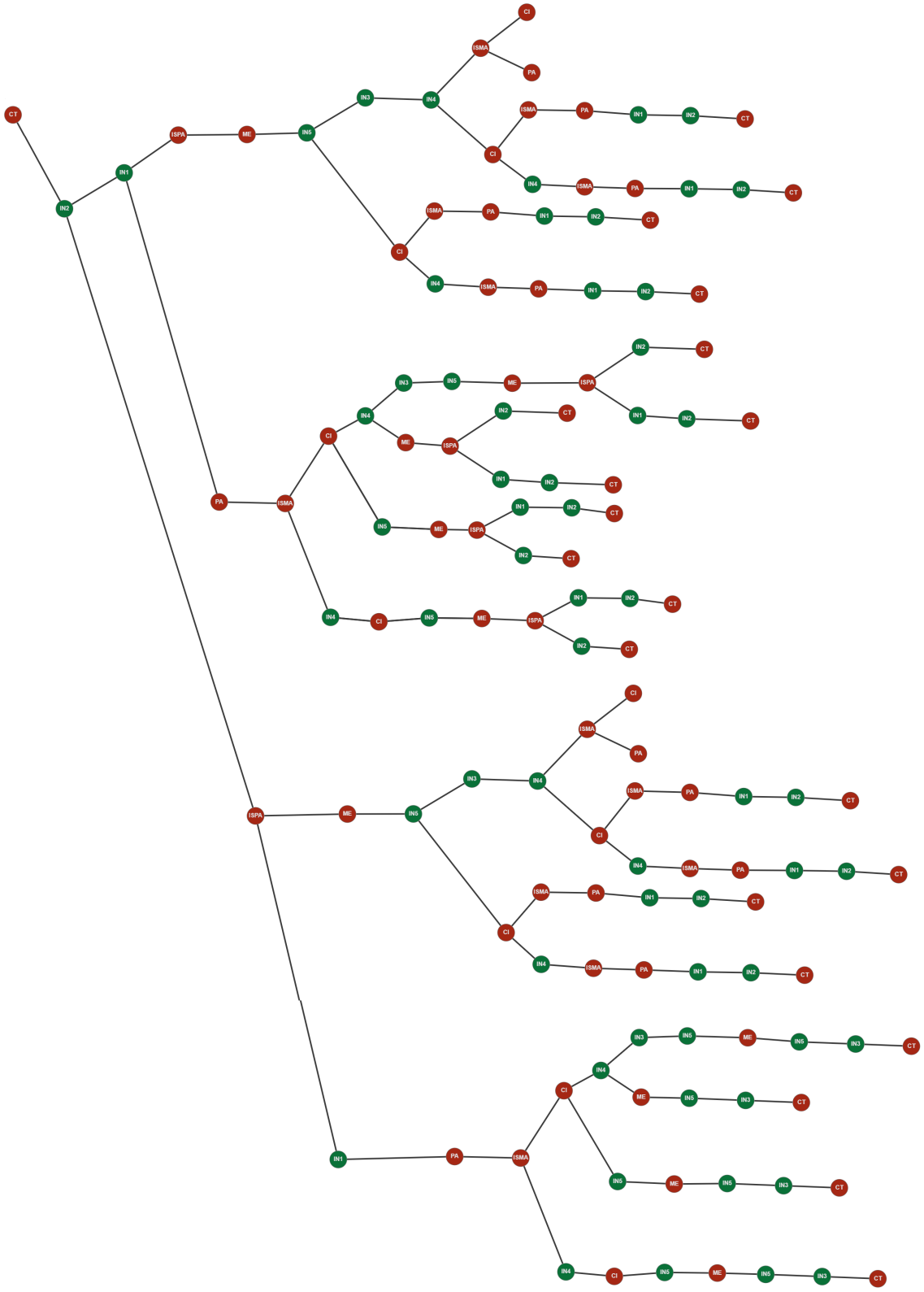
c) O percurso com menor custo com partida no alojamento e que permita visitar todos os 5 locais uma única vez e regressar ao local de partida)

Esta pergunta é muito parecida com o problema do Caixeiro Viajante. O problema do Caixeiro Viajante consiste em dado um conjunto de cidades e a distância entre cada par de cidades, encontrar a rota mais curta possível que visite todas as cidades exatamente uma única vez e retorne ao ponto de partida.
Para resolver o problema do Caixeiro Viajante existem diversos algoritmos como:

- Algoritmo força bruta que consiste em gerar as várias soluções possíveis e verificar a que satisfaz a condição do problema (percurso com menor custo).
- Algoritmo Branch and bound que consiste em subdividir o problema em problemas mais pequenos, de forma a obter a melhor solução.
- Algoritmo do Vizinho Mais Próximo que consiste em escolher um ponto de partida e de entre as arestas adjacentes escolher a que tem menor peso partindo para esse vértice a que chamamos vizinho mais próximo.

Para a resolução desta alínea utilizamos o algoritmo da força bruta pois é um dos mais fáceis de aplicar, apesar de não ser um dos mais eficientes é um dos mais exatos pois funciona como "tentativa e erro", tornando-se assim bastante preciso visto que a probabilidade de não obter o melhor caminho é muito ínfima.


IN1 PA ISMA IN4 IN3 CT
IN5 ME ISPA
IN2 IN1 PA ISMA IN4 IN3 CT
CI
IN2 ISPA ME IN5 IN3 CT
ISMA PA IN1
ISPA ME IN5 IN3 CT
IN4
IN2 IN1 PA IN1 IN2 CT
CI IN5 ME ISPA
IN1 PA IN1 IN2 CT
ISMA
IN4 IN3 CT
IN3 IN4 CI IN5 IN3 CT
PA IN1 IN2 ISPA ME IN5
CI IN4 IN3 CT
IN5 IN3 CT
ISPA ME IN5 IN3 IN4 CI IN4 IN3 CT
IN5 IN3 CT
CI IN4 IN3 CT
IN5 IN3 CT
IN2 ISPA ME IN5 IN3 CT
ISMA PA IN1
ISPA ME IN5 IN3 CT
CI
IN3 IN5 IN4 ISMA PA IN1 IN2 ISPA ME IN5 IN3 CT
ISPA ME IN5 IN3 CT
CT
ME IN2 IN1 PA ISMA CI IN4 IN3 CT
ISPA
IN5 IN3 CT
IN4 CI IN4 IN3 CT
IN5 IN3 CT
IN3 IN5 CI IN4 IN3 CT
IN5 IN3 CT
IN1 PA ISMA CI IN4 IN3 CT
IN5 IN3 CT
IN4 CI IN4 IN3 CT
IN4 IN5 IN3 CT
IN3 IN5 CI IN4 IN3 CT
IN5 IN3 CT


CT
IN2
IN1
ISPA
ME
IN5
IN3
IN4
CI
ISMA
PA
IN1
IN2
CT
CI
ISMA
PA
IN1
IN2
CT
IN4
ISMA
PA
IN1
IN2
CT
CI
ISMA
PA
IN1
IN2
CT
IN4
ISMA
PA
IN1
IN2
CT
PA
ISMA
CI
IN4
IN3
IN5
ME
ISPA
IN2
CT
IN1
IN2
CT
ME
ISPA
IN2
CT
IN1
IN2
CT
IN5
ME
ISPA
IN1
IN2
CT
IN2
CT
IN4
CI
IN5
ME
ISPA
IN1
IN2
CT
IN2
CT
ISPA
ME
IN5
IN3
IN4
CI
ISMA
PA
ISMA
PA
IN1
IN2
CT
CI
ISMA
PA
IN1
IN2
CT
IN4
ISMA
PA
IN1
IN2
CT
CI
ISMA
PA
IN1
IN2
CT
IN4
ISMA
PA
IN1
IN2
CT
IN1
PA
ISMA
CI
IN4
IN3
IN5
ME
IN5
IN3
CT
ME
IN5
IN3
CT
IN5
ME
IN5
IN3
CT
IN4
CI
IN5
ME
IN5
IN3
CT

Parte dos cálculos

CT IN3 IN4 CI IN5 ME ISPA IN1 PA ISMA IN4 IN3 CT = 14,73
CT IN3 IN4 CI IN5 ME ISPA IN2 IN1 PA ISMA IN4 IN3 CT = 14,58
CT IN3 IN4 CI ISMA PA IN1 IN2 ISPA ME IN5 IN3 CT = 16,08
CT IN2 ISPA ME IN5 IN3 IN4 CI ISMA PA IN1 IN2 CT = 12.26
CT IN2 ISPA ME IN5 CI ISMA PA IN1 IN2 CT = 8.78
CT IN3 IN5 CI ISMA PA IN1 IN2 ISPA ME IN5 IN3 CT = 17,06
CT IN3 IN5 CI ISMA PA IN1 ISPA ME IN5 IN3 CT = 16,67
CT IN2 IN1 ISPA ME IN5 IN3 IN4 CI ISMA PA IN1 IN2 CT = 12.51
CT IN2 IN1 ISPA ME IN5 IN3 IN4 CI IN4 ISMA PA IN1 IN2 CT = 13.18
CT IN2 IN1 ISPA ME IN5 CI ISMA PA IN1 IN2 CT = 10.39
CT IN2 IN1 ISPA ME IN5 CI IN4 ISMA PA IN1 IN2 CT = 11.06
CT IN2 IN1 PA ISMA CI IN4 IN3 IN5 ME ISPA IN2 CT = 12.26

...

Caminho mais curto: CT IN2 ISPA ME IN5 CI ISMA PA IN1 IN2 CT
Custo do caminho mais curto: 8.78

3. Exercício de Criptografía

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	.	,	!
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

3.1. Encriptação

Nesta questão é pedido para encriptar a mensagem "Venha experimentar, saborear e sentir. . . Felgueiras!" que é o slogan do Turismo de Felgueiras (https://visitfelgueiras.com/), acrescentando, "é o conselho de ... 4 primeiros nomes de cada um dos estudantes do grupo"através da seguinte função: $f(p) = (\beta p + 2) \mod 29$**, onde** $\beta \neq 0$ **e é o último algarismo do número de aluno de um dos elementos do grupo.**

Para β decidimos que seria $\beta = 6$, que vem do número mecanográfico 8200586, então:

$$f(p) = (\beta p + 2) \mod 29 = (6p + 2) \mod 29$$

Calculo auxiliar verificar se a função é bijetiva:
Para isso o mdc(6, 29) tem de ser igual a 1, ou seja, são primos entre si.
O máximo divisor comum são todos os fatores comuns elevados ao menor expoente.

$$\begin{array}{c|l} 6 & 2 \\ 3 & 3 \\ 1 & \end{array} \qquad \begin{array}{c|l} 29 & 29 \\ 1 & \end{array}$$

Após obtermos a função de encriptação, já podemos encriptar a frase. Para isso temos de ir à tabela e ver a que número corresponde cada letra da frase e substituir o p na função $f(p)$.

Frase original: Venha experimentar, saborear e sentir... Felgueiras! e o conselho de Bruno, Goncalo, Jorge e Nuno

$$\begin{aligned} &V = 21 \to f(21) = (6 \times 21 + 2) \mod 29 = 12 \to M \\ &e = 4 \to f(4) = (6 \times 4 + 2) \mod 29 = 26 \to . \\ &n = 13 \to f(13) = (6 \times 13 + 2) \mod 29 = 22 \to W \\ &h = 7 \to f(7) = (6 \times 7 + 2) \mod 29 = 15 \to P \\ &a = 0 \to f(0) = (6 \times 0 + 2) \mod 29 = 2 \to C \end{aligned}$$

Para tornar este processo mais rápido decidimos criar um programa em python para efetuar a encriptação da frase que queremos encriptar.

Este programa recebe uma frase, o número a ser multiplicado e o a ser somado, de seguida são efetuados os respetivos cálculos para cada caracter e após encriptar todos os carateres, é retornada a frase encriptada para o utilizador. Para verificar se é possível encriptar utilizamos um algoritmo já existente que se encontra no seguinte link https://www.rookieslab.com/posts/extended-euclid-algorithm-to-find-gcd-bezouts-coefficients-python-cpp-code.

Exemplo execução:

```
Valor a multiplicar:6
Valor a somar:2
Frase:Venha experimentar, saborear e sentir... Felgueiras! e o conselho de Bruno, Goncalo, Jorge e Nuno
Frase Encriptada: M.WPC .YF.RVQ.WACRT XCI!R.CR . X.WAVRNNN D.KJG.VRCXZ . ! O!WX.KP! U. IRGW!T J!WOCK!T ,!RJ. . WGW!
```

Tabela de encriptação:

Original	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	.	,	!
p	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
$f(p)$	2	8	14	20	26	3	9	15	21	27	4	10	16	22	28	5	11	17	23	0	6	12	18	24	1	7	13	19	25
Criptografada	C	I	O	U	.	D	J	P	V	,	E	K	Q	W	!	F	L	R	X	A	G	M	S	Y	B	H	N	T	Z

Frase original: Venha experimentar, saborear e sentir... Felgueiras! e o conselho de Bruno, Goncalo, Jorge e Nuno

Frase Criptografada: M.WPC .YF.RVQ.WACRT XCI!R.CR . X.WAVRNNN D.KJG.VRCXZ . ! O!WX.KP! U. IRGW!T J!WOCK!T ,!RJ. . WGW!

3.2. Desencriptação

Nesta questão é pedido para desencripte a mensagem obtida na alínea anterior.
A partir da função $f(p) = (6p+2) \mod 29$, que foi obtida na alínea anterior será feita a desencriptação.

Para fazer a desencriptação temos de encontrar uma expressão para $f^{-1}(x)$

$$\begin{aligned}
&(6p+2) \mod 29 = y \\
&(6p) \mod 29 + 2 \mod 29 = y \\
&(6p) \mod 29 = y - 2 \\
&6p \equiv (y-2) \mod 29
\end{aligned}$$

Calculo auxiliar do inverso de $6 \mod 29$:

Antes de calcular o inverso temos de verificar se é possível.
Para isso mdc(6, 19)=1 e como já visto em cima é verdade, assim sendo a função é invertível.

Agora temos de aplicar o algoritmo de Euclides para obtermos os coeficientes de Bézout:

$$a \times 6 \equiv 1 \mod 29$$

$$\begin{aligned}
29 = 6 \times 4 + 5 \rightarrow 5 &= 29 - 6 \times 4 \\
6 = 5 \times 1 + 1 \rightarrow 1 &= 6 - 5 \times 1 \\
&= 6 - (29 - 6 \times 4) \times 1 \\
&= 6 - 29 \times 1 + 6 \times 4 \\
&= -29 \times 1 + 6 \times 5
\end{aligned}$$

Por fim descobrimos que os coeficientes de Bézout são 1 e 5.

Agora temos de multiplicar a função pelo coeficientes de Bézout do número 6 que é 5.

$$6p \equiv (y-2) \mod 29 \quad (=) \quad \cancel{5 \times} 6p \equiv (5y-10) \mod 29$$

$\times 5$

Então a função da desencriptação é:

$$f^{-1}(y) = (5y - 10) \mod 29 = (5y + 19) \mod 29$$

Cálculo Auxiliar:
Como 29 divide $19 - (-10) = 29$, temos que $19 \equiv -10 \mod 29$.

$$M = 12 \to f(12) = (5 \times 12 + 19) \mod 29 = 21 \to V$$
$$. = 26 \to f(26) = (5 \times 26 + 19) \mod 29 = 4 \to E$$
$$W = 22 \to f(22) = (5 \times 22 + 19) \mod 29 = 13 \to N$$
$$P = 15 \to f(15) = (5 \times 15 + 19) \mod 29 = 7 \to H$$
$$C = 2 \to f(2) = (5 \times 2 + 19) \mod 29 = 0 \to A$$

Para tornar este processo mais rápido decidimos usar o programa da encriptação, usando os dados da função $f^{-1}(y)$.

Exemplo execução:

```
Valor a multiplicar:5
Valor a somar:19
Frase:M.WPC .YF.RVQ.WACRT XCI!R.CR . X.WAVRNNN D.KJG.VRCXZ . ! O!WX.KP! U. IRGW!T J!WOCK!T ,!RJ. . WGW!
Frase Encriptada: VENHA EXPERIMENTAR, SABOREAR E SENTIR... FELGUEIRAS! E O CONSELHO DE BRUNO, GONCALO, JORGE E NUNO
```

Tabela de desencriptação:

Criptografada	C	I	O	U	.	D	J	P	V	,	E	K	Q	W	!	F	L	R	X	A	G	M	S	Y	B	H	N	T	Z
y	2	8	14	20	26	3	9	15	21	27	4	10	16	22	28	5	11	17	23	0	6	12	18	24	1	7	13	19	25
$f^{-1}(y)$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Original	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	.	,	!

Frase criptografada: M.WPC .YF.RVQ.WACRT XCI!R.CR . X.WAVRNNN D.KJG.VRCXZ . ! O!WX.KP! U. IRGW!T J!WOCK!T ,!RJ. . WGW!

Frase descriptografada: VENHA EXPERIMENTAR, SABOREAR E SENTIR... FELGUEIRAS! E O CONSELHO DE BRUNO, GONCALO, JORGE E NUNO

Para tornar o processo da desencriptação mais fácil e rápido decidimos criar um programa em python.

Este programa recebe uma frase encriptada, o número a ser multiplicado e o a ser somado na encriptação, e de seguida são efetua os respetivos cálculos para cada carater e após encriptar todos os carateres, é retornada a frase desencriptada para o utilizador. Para descobrir os coeficientes de Bézout e o mdc utilizamos um algoritmo já existente que se encontra no seguinte link https://www.rookieslab.com/posts/extended-euclid-algorithm-to-find-gcd-bezouts-coefficients-python-cpp-code

Exemplo execução:

```
Valor a multiplicar na encriptacao:6
Valor a somar na encriptacao:2
Frase Encriptada:M.WPC .YF.RVQ.WACRT XCI!R.CR . X.WAVRNNN D.KJG.VRCXZ . ! O!WX.KP! U. IRGW!T J!WOCK!T ,!RJ. . WGW!
Frase Desencriptada: VENHA EXPERIMENTAR, SABOREAR E SENTIR... FELGUEIRAS! E O CONSELHO DE BRUNO, GONCALO, JORGE E NUNO
```

Conclusão

Ao longo deste trabalho surgiram algumas dúvidas que foram esclarecidas pelas professoras responsáveis da Unidade Curricular e através de algumas pesquisas na Internet. Na nossa opinião achamos que o trabalho foi concluído com sucesso sendo que solucionamos todos os exercícios com sucesso explicando sempre as nossas resoluções e apresentando sempre outras possíveis soluções.

Tal como já foi referido na introdução deste relatório, todos os conceitos abordados serão bastante importantes para o nosso futuro sendo que temos vários exemplos em que estes conhecimentos podem ser utilizados, como por exemplo:

Exemplo do uso da recursividade:
A recursividade é bastante utilizada na programação quando uma parte do algoritmo necessita do resultado da última iteração, este processo permite obter uma maior legibilidade do código bem como uma melhoria em termos de processamento.

Exemplo de uso da Teoria de Grafos:
Temos como exemplo uma estrutura de ligação de um Website de artigos onde os vértices podem ser os artigos em questão e as arestas as ligações entre os diversos artigos, ou seja, imaginando que temos o artigo A e o artigo B se existir uma aresta entre estes dois vértices significa que existe por exemplo um link do artigo A para o B.

Exemplo de encriptação e desencriptação:
A encriptação e desencriptação são conceitos muito importantes sendo que, num contexto empresarial, existe sempre informação confidencial que tem de ser protegida e, para isso é então utilizada a encriptação para que esta informação não seja visualizada por qualquer um e apenas por pessoas que tenham acesso ao algoritmo de desencriptação.

Concluímos assim que este trabalho além de ser útil para demonstrarmos os nossos conhecimentos, foi também muito importante para a aprendizagem de novos conceitos relacionados com a matéria.

Bibliografia

[1] *Bézout coeficientes e mdc python*. https://www.rookieslab.com/posts/extended-euclid-algorithm-to-find-gcd-bezouts-coefficients-python-cpp-code.

[2] *Desenhador de grafos*. https://www.graphonline.ru/pt/.

[3] *EGV exercício*. https://www.chegg.com/homework-help/questions-and-answers/f-1-2-f-n-2f-n-1-2-n-n-2-use-expand-guess-verify-prove-answer-mathematical-induction-q39051578?trackid=26ba2bbae6f4&strackid=5a826b579942.

[4] *Gerador de tablas Latex*. https://www.tablesgenerator.com/.

[5] *GraphTea*. http://www.graphtheorysoftware.com.

[6] Tecnico Lisboa. *Método de indução matemática*. http://e-escola.tecnico.ulisboa.pt/topico.asp?id=5&ordem=5.

[7] *Overleaf*. https://www.overleaf.com.

[8] *Rotas Românticas em Portugal*. https://www.rotadoromanico.com/pt.

[9] *Scilab*. https://www.scilab.org.

[10] Alan De Smet. *Algoritmo de Dijkstra*. https://gitlab.com/AlanDeSmet/dijkstra-spf/-/tree/master.

[11] *Waze*. https://www.waze.com/pt-PT/.