AED

Algoritmos e Estrutura de Dados

Traveling Salesman Problem

11/11/2018

Bernardo Rodrigues nº Mec: 88835

**Introdução**

Para a disciplina de Algoritmos e Estrutura de Dados foi-nos proposto a resolução do “Travelling Salesman Problem” um problema da teoria de grafos que envolve o cálculo do melhor e pior caminho que um viajante pode percorrer quando quer viajar por um determinado número de cidades, regressando de volta à cidade inicial.

TSP

Inicialmente para resolver este problema escrevemos uma função que calcula todas as permutações possíveis de fazer com n cidades, mantendo constante a cidade inicial, visto que vamos regressar à cidade inicial no final do tour, e em cada permutação calcula a distância necessária para percorrer as cidades.

No final compara a distância total calculada com a distância mínima e máxima que tem registadas e atualiza conforme necessário, guardando também o percurso para a distancia

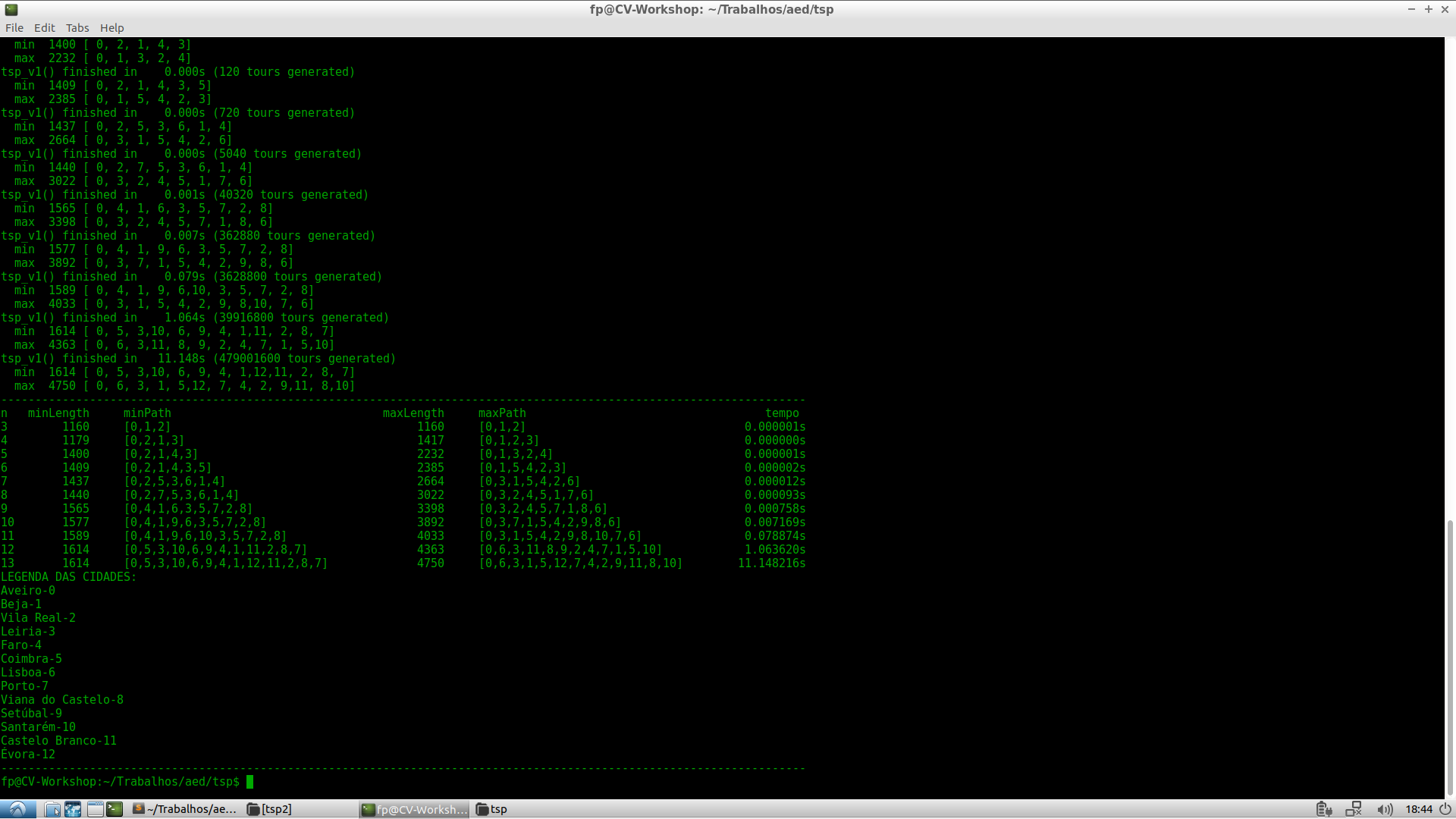
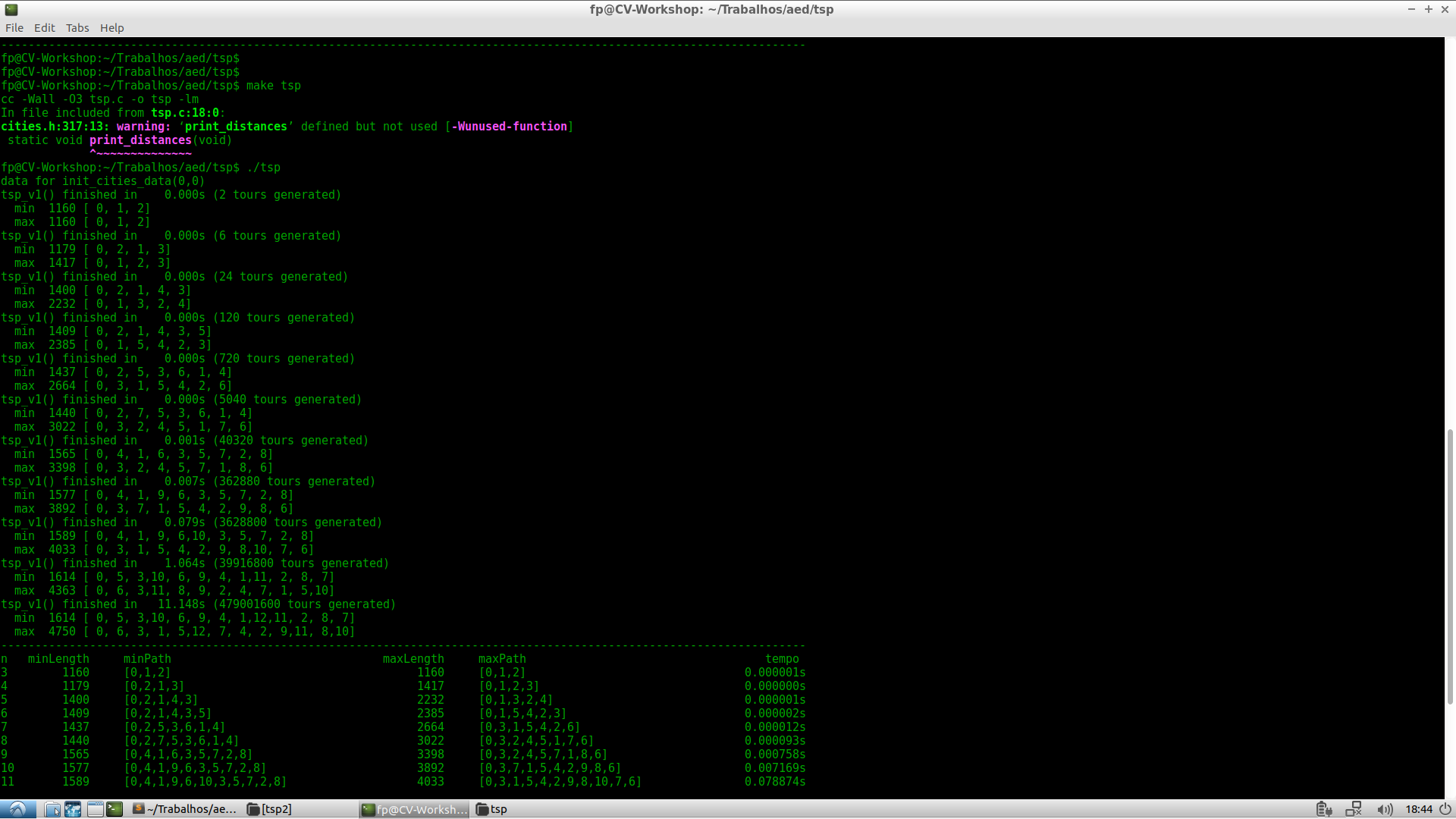
O programa vai repetindo os cálculos começando por realizar tours com apenas 3 cidades, até ao máximo de cidades (18), começando sempre os cálculos de novo cada vez que é adicionada uma nova cidade

No final o programa dá a indicação do melhor e pior caminho para cada número de cidades percorridas e o respetivo percurso para o atingir

Como foi preciso calcular todos os possíveis percursos foram necessários o cálculo de n! caminhos diferentes, sendo n o número de cidades por onde o viajante passou.

Isto dá ao problema uma complexidade de O(n!) o que significa que para um número elevado de cidades o programa teria um tempo de execução enorme.

Gráfico apenas para 14 cidades, porém o problema contém 18

Prints do resultado do programa

Esta abordagem apesar de simples, não é eficiente pois demora demasiado tempo, nomeadamente porque repete várias vezes cálculos que já foram previamente realizados,

por exemplo: quando calcula a viagem pelas cidades 0-1-2-3-4-5, vai repetir desnecessariamente parte dos cálculos para a combinação das cidades (0-1-2-3) (5-4) e é esta repetição de cálculos que aumenta imenso o tempo do programa

TSP2

Para tornar o programa mais eficaz, recorremos ao uso de dynamic programing e de memoization

Deste modo o programa calcula recursivamente as distâncias percorridas, dividindo o problema em problemas menores, ou seja, para calcular a rota para 6 cidades o programa vai primeiro calcular as rotas para 5, que por sua vez vai calcular as rotas para 4 cidades, chegando ao mínimo de 3 cidades.

Adicionalmente a cada posição é atribuída uma mascara, isto é por cada posição visitada é posto um bit de uma variável a 1, por exemplo quando temos 4 cidades e é visitada a cidade 0 (sempre a primeira cidade a ser visitada) temos a mascara 0001, se a seguir visitarmos a cidade 2 o terceiro bit da mascara (a contar da direita) é posto a 1, ficando a mascara 0101, sendo o percurso feito até a momento 0-2-…

O programa inicia o tour na cidade 0 e a mascara a 1, vai chamando a função recursivamente ponde sempre um bit da mascara a 1, até que a mascara tenha o valor 2n-1 que é o valor máximo que pode ter quando todos os bits estão a 1 (exemplificando quando temos 4 cidades a mascara cheia ficaria 11112 ou seja 24-1 = 15 ), e retorna a distancia da cidade onde estamos correntemente à cidade inicial.

O programa vai repetindo-se e guarda os valores dos sub-tours num array de array (memoization). O array vai ter capacidade [2n][n] onde será guardada a distância de percursos previamente percorridos, tais percursos serão guardados na posição do array [mascara][cidade atual], deste não estaremos que estar a repetir cálculos pois já sabemos a distância.

Exemplificado:

Se tivermos 5 cidades e calcularmos a distancia de 3-2-1, guardamos o valor na posição [001111] [3] (nota: o primeiro bit vai estar sempre a 1) assim quando calcularmos o percurso 4-3-2-1 teremos apenas de somar a distancia de 4 ao valor guardado na posição [001111] [3], o que vai reduzir imenso o tempo de execução do programa, visto que o numero de cálculos a ser realizado vai diminuir.

Porém para guardar os valores dos percursos vai ser preciso usar alguma memoria do computador.