Universidade do Minho
Escola de Engenharia

Licenciatura em Engenharia Informática
Investigação Operacional

Trabalho Prático 2

Jéssica Cunha (a100901)
Martim Redondo(a100664)
Rodrigo Castro (a100694)
Tiago Moreira (a100541)

Ano letivo 2022/23

Índice

Introdução

Este trabalho de IO visa a matéria de problemas de transporte, no âmbito deste trabalho prático há alguns passos que tivemos de seguir para resolver o problema, primeiro tivemos que transformar o enunciado tendo em conta o elemento do nosso grupo com maior número, de seguida teríamos que organizar essa informação tendo em conta alguns dados. Posto isto, fazíamos um grafo de compatibilidade, transformámo-lo em um ficheiro que será executado com o relax4 onde obteríamos o output e a resposta ao nosso problema.

Problema

Para conseguirmos resolver o problema teríamos que, primeiro, pegar no maior número do elemento do nosso grupo, que é 100901. Depois de dividi-lo por letras chegamos às seguintes letras:

A=0;
B=0;
C=9;
D=0;
E=1;

O que nos levou a concluir a tabela, que antes estaria incompleta, pois os valores em falta eram a1 e a8 que agora são:
a1 = 1 ;
a8 = 10;
Chegamos à conclusão tammbém que teríamos que excluir o cliente D, mas **unicamente quando formos criar o ficheiro relax4.**

Tabela mudada:

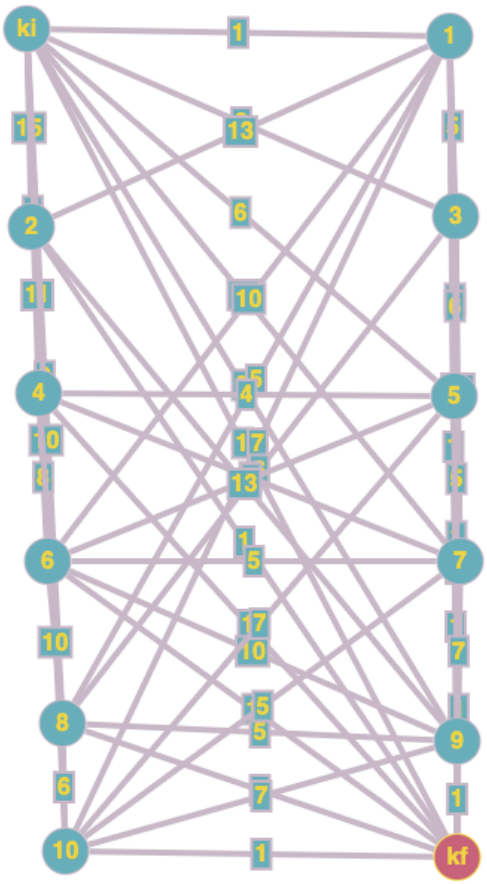
j	cliente	a_j (¼hora)	a_j (hora do serviço)
1	Ana	1	09:15
2	Beatriz	7	10:45
3	Carlos	4	10:00
4	Diogo	2	09:30
5	Eduardo	10	11:30
6	Francisca	6	10:30
7	Gonçalo	9	11:15
8	Helena	10	11:30
9	Inês	2	09:30
10	José	5	10:15

Grafo de Compatibilidades

Depois da tabela de dados bem definida passamos para a criação do Grafo de compatibilidades que nos facilitaria a resolução do trabalho prático. Para tal entramos num site própria para esse efeito e utilizando uma tabela de edge list criamos o nosso gráfico.

Link: http://graphonline.ru/pt/?graph=gsirfHreiYkJcHZj

Grafo de compatibilidade:



Sabemos que o gráfico não está o mais explícito, por isso fornecemos o link e ainda vamos adicionar a “tabela” de edge list que usamos para construir este grafo:

ki-(1)-1
ki-(15)-2
ki-(2)-3
ki-(4)-4
ki-(6)-5
ki-(11)-6
ki-(11)-7
ki-(9)-8
ki-(9)-9
ki-(10)-10
ki-(1)-kf
1-(13)-2
1-(5)-3
1-(5)-5
1-(10)-6
1-(7)-7
1-(5)-8
1-(0)-9
1-(7)-10
1-(1)-kf
2-(1)-kf

3-(6)-5
3-(10)-7
3-(6)-8
3-(1)-kf
4-(4)-5
4-(8)-6
4-(8)-7
4-(8)-8
4-(1)-kf
5-(1)-kf
6-(6)-5
6-(5)-7
6-(10)-8
6-(1)-kf
7-(1)-kf
8-(1)-kf
9-(13)-2
9-(5)-3
9-(5)-5
9-(10)-6
9-(7)-7
9-(5)-8
9-(7)-10
9-(1)-kf
10-(7)-5
10-(5)-7
10-(6)-8
10-(1)-kf

Ficheiro Input e OutPut:

Para fazer o ficheiro input tivemos que ter algumas coisas em consideração. A primeira delas é que como queremos que todos os clientes sejam atendidos precisamos de induzir o relax4 a tomar esse caminho e para tal temos que dividir os clientes em "cliente destino" e "cliente partida", onde o cliente destino teria uma procura de 1 (logo valor é de –1) e o cliente partida teria uma oferta de 1 (logo valor é de 1).

Outro fator a ter em conta é a forma que arranjamos de "remover" um cliente, que foi atribuir-lhe um valor 0, pois assim estaríamos a dar-lhe uma capacidade residual de 0, obrigando o relax4 a ignorar tal cliente.

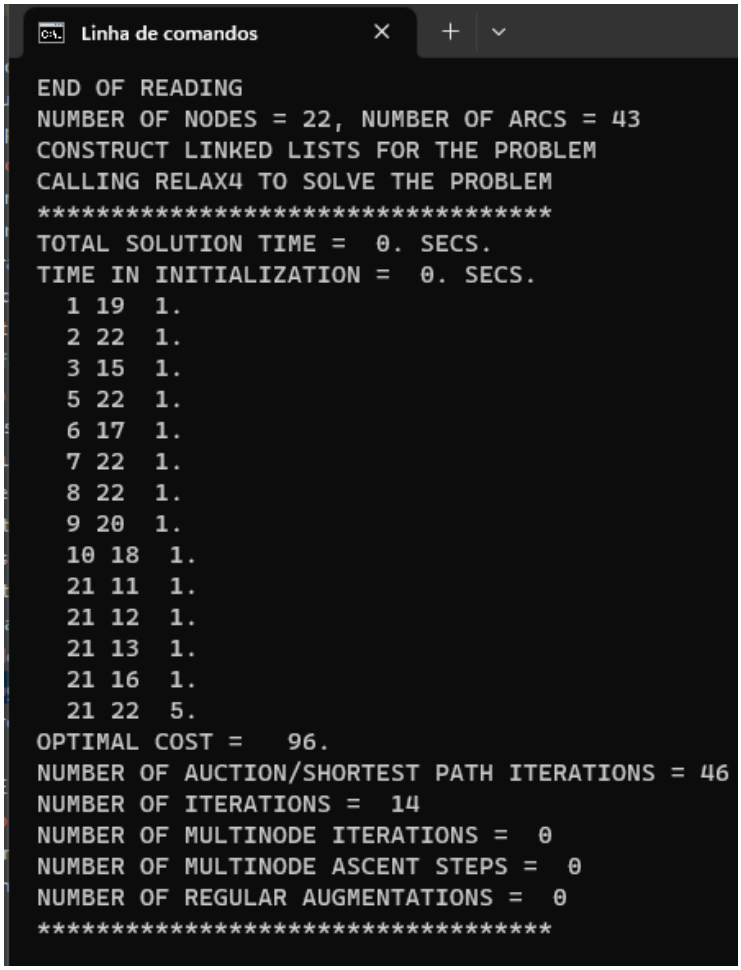
Para concluir decidimos criar um arco direto da sede para a sede para conseguirmos ver que todos os clientes estavam atendidos e assim contabilizar quantas equipas eram necessárias.

Input:

```
22
43
1 12 13 1000
1 13 5 1000
1 15 5 1000
1 16 10 1000
1 17 7 1000
1 18 5 1000
1 19 0 1000
1 20 7 1000
1 22 2 1000
2 22 16 1000
3 15 6 1000
3 17 10 1000
3 18 6 1000
3 22 3 1000
5 22 7 1000
6 15 6 1000
6 17 5 1000
6 18 10 1000
6 22 12 1000
7 22 10 1000
8 22 10 1000
9 12 13 1000
9 13 5 1000
9 15 5 1000
9 16 10 1000
9 17 7 1000
9 18 5 1000
9 20 7 1000
9 22 10 1000
10 15 7 1000
10 17 5 1000
10 18 6 1000
10 22 11 1000
```

```
21 11 1 1000
21 12 15 1000
21 13 2 1000
21 15 6 1000
21 16 11 1000
21 17 9 1000
21 18 9 1000
21 19 9 1000
21 20 10 1000
21 22 0 1000
1
1
1
0
1
1
1
1
1
1
-1
-1
-1
0
-1
-1
-1
-1
-1
-1
9
-9
```

OutPut:



Interpretação dos resultados:

Como podemos ver pelo output o menor custo de deslocamento que podemos ter é 96, sendo que os caminhos escolhidos são:

Sede ->Cliente1->Cliente9 ->Cliente10->Cliente8->Sede;

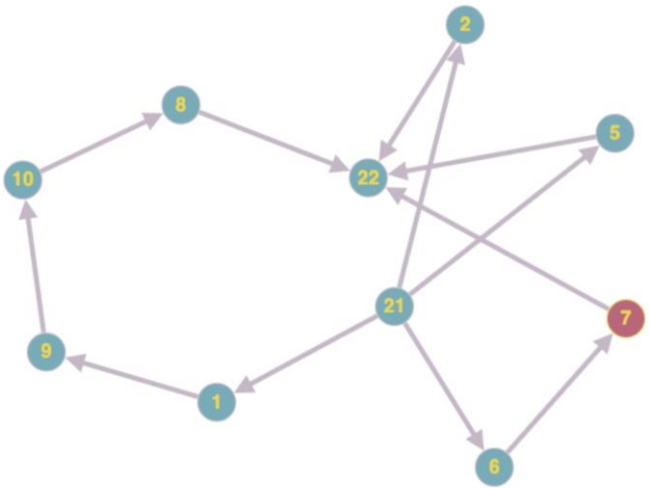
Sede->Cliente2->Sede;

Sede->Cliente5->Sede;

Sede->Cliente6->Cliente7->Sede;

Sede->Sede.

Grafo que representa o caminho seguido pelas equipas:



Como podemos ver são precisas exatamente 4 equipas para suprir as necessidades de todos os clientes, a quinta equipa sai e volta para a sede porque todos os clientes já se encontram todos atendidos.