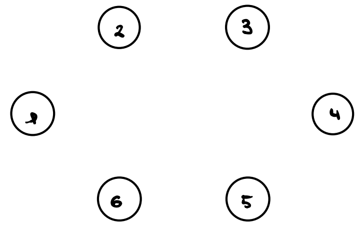
	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
4	30	9	9		8	7
5	8	7	30	8		7
6	12	13	16	7	7	



Heurística nearest neighbour (TSP)

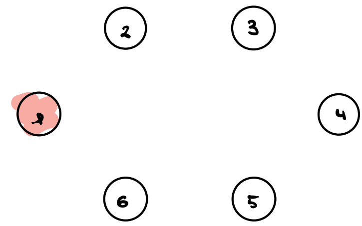
Nearest Neighbor Heuristic

1. Select a node i∈N, label it, and make T:=∅ and p:=i
2. If all nodes are labeled make T:=T∪{i_p }. STOP, T is a Hamiltonian circuit
3. Select a not yet labeled node j such that: c_{pj} = MIN {c_{pk} |k not labeled}
4. Make:
   T:=T∪{c_{pj} }
   Label j
   Let p:=j
   Repeat from 2

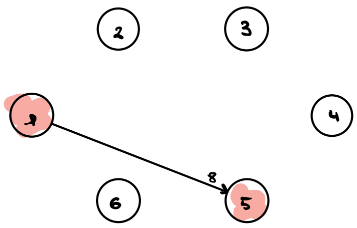
1. Seleccionamos 1

T = ∅ p = {1} label 1.

	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
4	30	9	9		8	7
5	8	7	30	8		7
6	12	13	16	7	7	

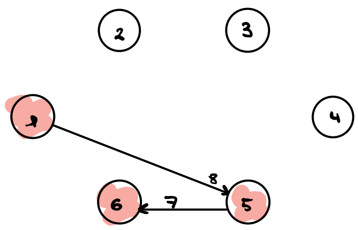


2. Next
3. min de los adj → 5
4. T = {8}
   label 5
   p = {5}



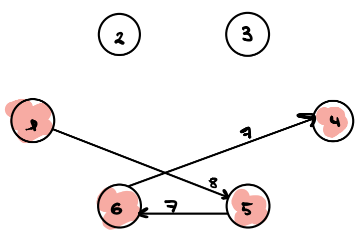
	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
4	30	9	9		8	7
5	8	7	30	8		7
6	12	13	16	7	7	

2. Next.
3. min → 2 o 6 → random: 6
4. T = {8,7}
   label 6
   p = {6}



	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
4	30	9	9		8	7
5	8	7	30	8		7
6	12	13	16	7	7	

2. Next.
3. min → 4
4. T = {8,7,7}
   label 4
   p = {4}



	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
4	30	9	9		8	7
5	8	7	30	8		7
6	12	13	16	7	7	

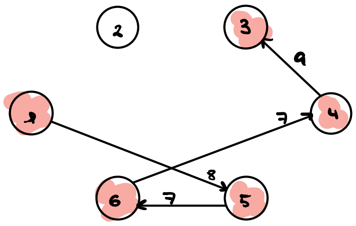
2. Next.

3. min → 3

4. T = {8,7,7,9}

label 3

P = {3}



	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
4	30	9	9		8	7
5	8	7	30	8		7
6	12	13	16	7	7	

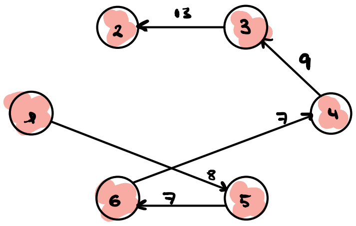
2. Next.

3. min → 2

4. T = {8,7,7,9,13}

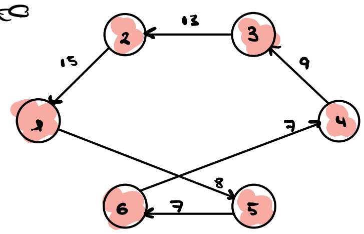
label

P = {2}



	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
4	30	9	9		8	7
5	8	7	30	8		7
6	12	13	16	7	7	

final → ajuste



Algoritmo de Prim para MST (TSP)

1. **Select** an arbitrary node w∈V, **make** ST:= ∅ , W:={w}, V:=V \ W
2. **If** V=∅ END. ST is a "Minimum Spanning Tree"
3. **Select** an arc (u,v) ∈ A with u ∈ W and v ∈V such that:

   c_{uv} = Min { c_e | e ∈ δ (W) } (δ (W) = {set of arcs of A with a node in W and other in V })

4. **Make**:

   ST := ST ∪ {(u,v)}

   W := W ∪ {v}

   V := V \ {v}

Go to 2.

1. W = 4

   ST = ∅

   W = {4}

   V = {1,2,3,5,6}

	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
→ 4	30	9	9		8	7
5	8	7	30	8		7
6	12	13	16	7	7	

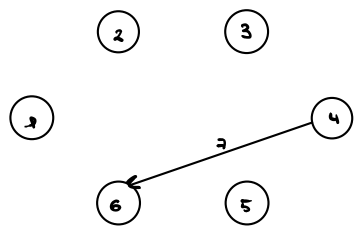
2. Next

3. 4 → 6

4. ST = {(4,6)}

   W = {4,6}

   V = {1,2,3,5}



2. Next

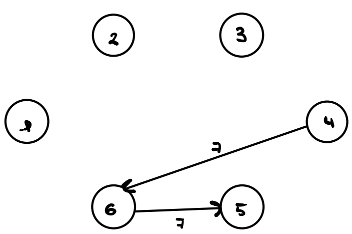
3. 6 → 5

4. ST = {(4,6)(6,5)}

   W = {4,6,5}

   V = {1,2,3}

	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
→ 4	30	9	9		8	7
5	8	7	30	8		7
→ 6	12	13	16	7	7	



2. Next

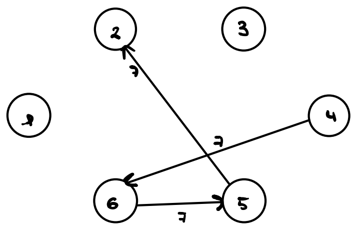
3. 5 → 2

4. ST = {(4,6)(6,5),(5,2)}

W = {4,6,5,2

V = {1,3}

	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
→ 4	30	9	9		8	7
→ 5	8	7	30	8		7
→ 6	12	13	16	7	7	



2. Next

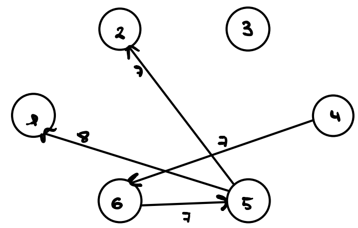
3. 5 → 1

4. ST = {(4,6)(6,5),(5,2)(5,1)}

W = {4,6,5,2,1}

V = {3}

	1	2	3	4	5	6
1		15	30	30	8	12
→ 2	15		13	9	7	13
3	30	13		9	30	16
→ 4	30	9	9		8	7
→ 5	8	7	30	8		7
→ 6	12	13	16	7	7	



2. Next

3. 4 → 3

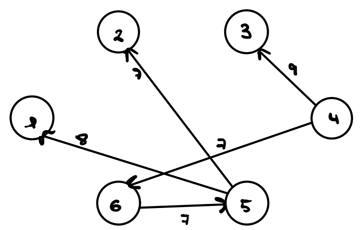
4. ST = {(4,6)(6,5),(5,2)(5,1)} (4,3)

W = {4,6,5,2,1,3

V = ∅

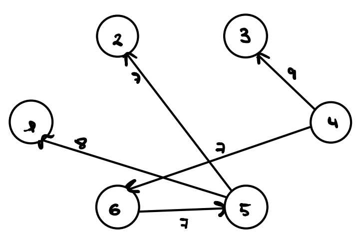
	1	2	3	4	5	6
→ 1		15	30	30	8	12
→ 2	15		13	9	7	13
3	30	13		9	30	16
→ 4	30	9	9		8	7
→ 5	8	7	30	8		7
→ 6	12	13	16	7	7	

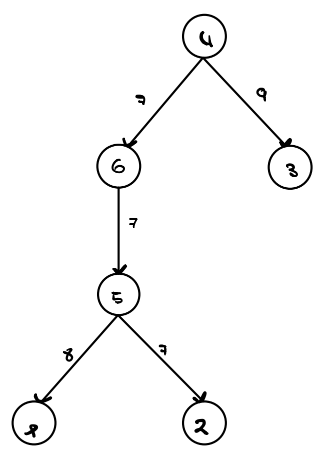
final.

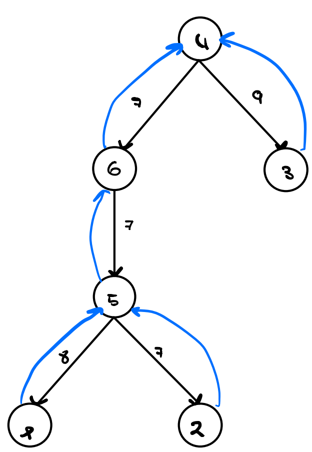


Heurística spanning tree (TSP)

1. **Find** a "Minimum Spanning Tree" ST of K_n.
2. **Double** all arcs of ST to generate the auxiliary graph (V,ST)
3. Since all nodes of (V,ST) have even degree and (V,ST) is connex, (V,ST) is Eulerian.
   - **Determine** an eulerian circuit C in (V,ST)
   - **Assign** an orientation to C
   - **Select** a node $i \in V$, **label it** and **make** p:=i, T:=∅
4. **If** all nodes are labeled **make** $T:=T \cup \{(i,p)\}$. END. T is a hamiltonian circuit .
5. **Otherwise, move** from p, along C, following the orientation in C until an unlabeled node q.
   Make $T:=T \cup \{(p,q)\}$. **Label** q, **Make** p:=q, **Go to 4**.





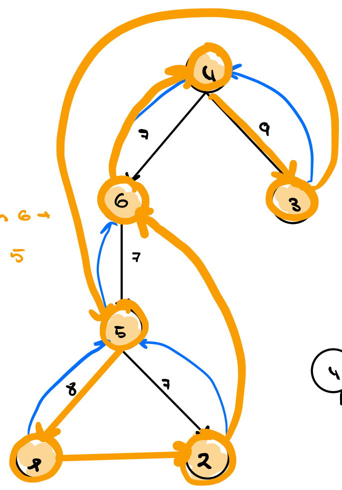


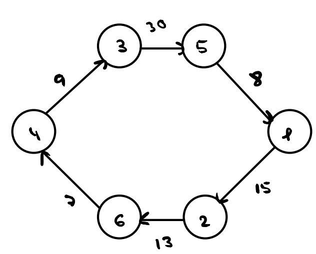
Double arcs.

Circuito euleriano

5 → 1 → 5 → 2 → 5 → 6 →

→ 4 → 3 → 4 → 6 → 5





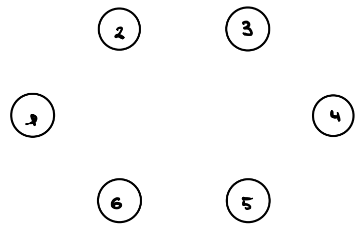
Heurística Clarck and Wright (VRP)

THE CLARKE AND WRIGHT:Savings heuristic

depot 4

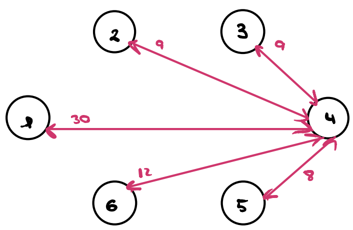
- **Assuming a graph G=(N,A), with a set of nodes N={0, 1, ..., n}, with the central depot at node 0, and customers or demand nodes at 1, ...n; and a set of arcs A=NxN. Begin with an infeasible solution in which every customer is supplied individually by a separate vehicle.**
- **Combine any two of these single customer routes to use one less vehicle and reduce cost:**
  - The cost of servicing customers i and j individually by two vehicles is $c_{0i}+c_{i0}+c_{0j}+c_{j0}$
  - The cost of one vehicle serving i and j on the same route is $c_{0i}+c_{ij}+c_{j0}$
  - Combining i and j results in cost saving of $s_{ij}= c_{i0}+c_{0j}-c_{ij}$
- **Select the arc (i,j) with maximum saving s_{ij} subject to the requirement that the combined route is feasible (i.e does not exceed vehicle capacity).**
- **Repeat the process until the number of routes is reduced to** $K=\left\lceil \frac{\sum_{i=1}^{n} a_i}{b} \right\rceil$

	1	2	3	4	5	6
1		15	30	30	8	12
2	15		13	9	7	13
3	30	13		9	30	16
4	30	9	9		8	7
5	8	7	30	8		7
6	12	13	16	7	7	



Definimos demandas $a = \{ 1, 2, 5, 7, -, 6, 3 \}$ (índices 1, 2, 3, 4, 5, 6) ↑ oferta $\rightsquigarrow K = \left\lceil \frac{23}{15} \right\rceil = 2$

Definimos capacidad $C = 15$



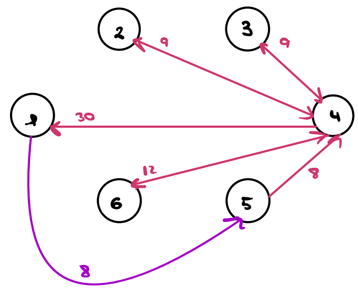
1 para cada cliente

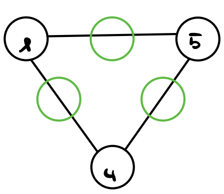
$s_{ij} = c_{i4} + c_{4j} - c_{ij}$

i=1	j=2	30	+	9	-	15	=	24
	j=3	30	+	9	-	30	=	9
	j=5	30	+	8	-	8	=	30
	j=6	30	+	7	-	12	=	25
i=2	j=3	9	+	9	-	13	=	5
	j=5	9	+	8	-	7	=	10
	j=6	9	+	7	-	13	=	3
i=3	j=5	9	+	8	-	30	=	-13
	j=6	9	+	7	-	16	=	0
i=5	j=6	8	+	7	-	7	=	8

⇝ para todos $d_i + d_j \leq C$ ⇝ no hace falta mirarla

→ max = 30 ⇝ (i, j) = (1, 5)





Para 2 ⇝ demanda = 13 ≤ C = 15 ✓

4 2 1 5 4 ⇝ 9+15+8+8 = 40

4 1 2 5 4 ⇝ 30+15+7+8 = 60

4 1 5 2 4 ⇝ 30+8+7+9 = 54

Para 3 ⇝ 15 ≤ 15 = C ✓

4 3 1 5 4 ⇝ 9+30+8+8 = 55

4 1 3 5 4 ⇝ 30+30+30+8 = 98

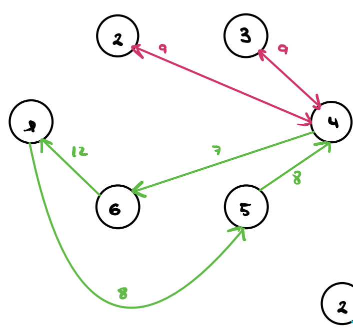
4 1 5 3 4 ⇝ 30+8+30+9 = 77

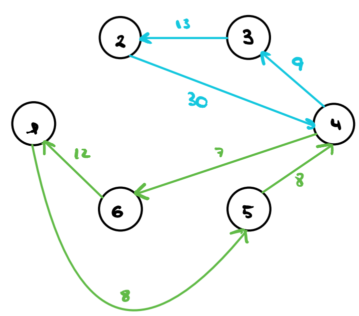
Para 6 ⇝ demanda 11 ≤ 15 = 6 ✓

4 6 1 5 4 ⇝ 7+12+8+8 = 35

4 1 6 5 4 ⇝ 30+12+7+8 = 57

4 1 5 6 4 ⇝ 30+8+7+7 = 52



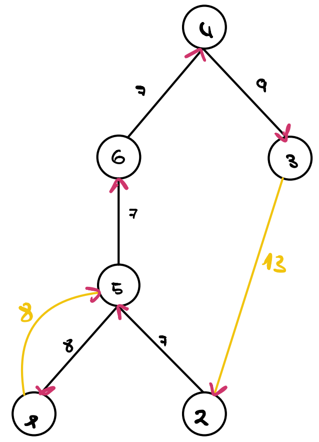


Si intentem posar 2 e 3 ⇝ demanda > C ⇝ en una altra ruta

Heurística de Cristofides

CHRISTOFIDES HEURISTIC

1. **Determine** a Minimum Spanning Tree ST of K_n.
2. Let W be the set of vertices with odd degree in ST. **Find** a Minimum Perfect Matching M in the complete graph K_n over the vertices from M
3. **Combine** the edges of M and ST to form a multigraph H.
4. Form an Eulerian Circuit in H (H is Eulerian because it is connected with only even degree vertices). **Give** H an orientation (arbitrary)
5. Select a node **i∈H,** label it and **make p:=i, T:=∅**
6. **If** all nodes are labeled then **make T:=T∪{(i,p)}. END. T is a hamiltonian circuit.**
7. **Otherwise ,** move from p, along H following the orientation, until the first unlabeled node **q. Make T:=T∪{(p,q)}. Label q, make p:=q, Repeat from ~~1~~. 6**



2. W = {1, 2, 3, 5}

Minimum perfect matching

(1,2) – (3,5) → 15 + 30 = 45
(1,3) – (2,5) → 30 + 7 = 37
(1,5) – (2,3) → 8 + 13 = 21

4. orientation

5. select i → 6 and label it

p = 6

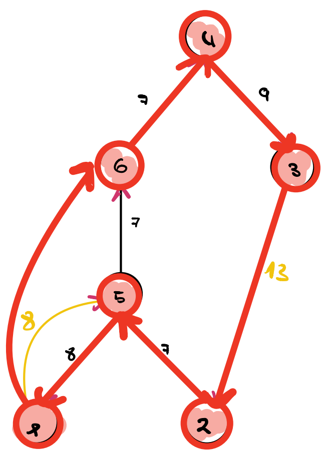
T = ∅

6. next

7. 6 → 4

T = {(6,4)}

6. como spanning tree



Heurística d'intercanvi 2 a 2

2-EXCHANGE HEURISTIC

<u>Procedure 2-opt</u>.

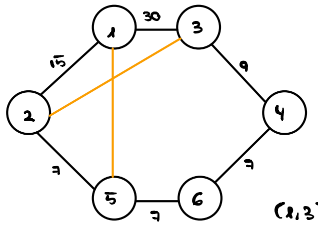
1. **Let** T be the current hamiltonian circuit.
2. **While** it is still possible for each node i, **do** the following exchange :
   2.1 **Select** a node i
   2.2 **Examine** all 2-OPT MOVE that consider the between i and its successor in the circuit . **If** it is possible to shorten the length of the circuit in this manner, **then select** the best of such 2-OPT MOVE, **otherwise declare** failed the process for node i.
3. **Repeat** from 1.

<u>Example:</u>

1. **Let** $T = \{i_1, i_2, \ldots, i_n\}$ **be** the current hamiltonian circuit. **Give** a circulation sense.
2. **Make** $Z := \{ (i_p, i_{p+1}), (i_q, i_{q+1}) \mid p+1 \neq q, p \neq q, q+1 \neq p, 1 \leq p, q \leq n \}$
3. For all pairs of arcs $\{(i_p, i_{p+1}), (i_q, i_{q+1})\} \in Z$ **make**:

$$\text{If} \quad c_{pp+1} + c_{qq+1} > c_{i_p i_q} + c_{i_{p+1} i_{q+1}} \quad \text{then make:}$$
$$T := \left(T \setminus \{(i_p, i_{p+1}), (i_q, i_{q+1})\}\right) \cup \{(i_p, i_q), (i_{p+1}, i_{q+1})\}$$

4. **Go to 2.**



cost = 30 + 9 + 7 + 7 + 7 + 15 = 75

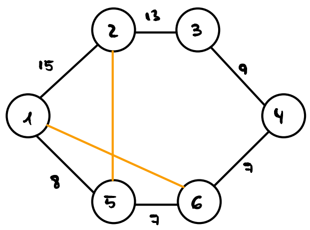
Inicialment ⇝ aresta de cost màx (1, 3) ← 30

- Provar d'intercanviar amb arestes NO contigües. {(4,6), (6,5), (5,2)}

(1,3)(4,6) = 30 + 7 = 37
- ⇝ (1,4)(3,6) = 30 + 16 = 46
- ⇝ (1,6)(3,4) ⇝ NO es POT

(1,3)(6,5) = 30 + 7 = 37
- ⇝ (1,6)(3,5) = 12 + 30 = 42
- ⇝ (1,5)(3,6) → no es pot

(1,3)(5,2)
- ⇝ (1,5)(3,2) = 8 + 13 = 21
- ⇝ (1,2)(3,5) ⇝ no es pot



(1, 2)
- 24 (3,4) ⇝ (1,3)(2,4) = 30 + 9 = 39
- 22 (4,6) ⇝ (1,4)(2,6) = 30 + 13 = 43
- 22 (6,5) ⇝ (1,6)(2,5) = 12 + 7 = 19