Annex A

Programari

Per poder realitzar aquest treball ha estat necessari fer un ús extensiu d'eines informàtiques. Penso que és important posar en valor totes aquestes eines utilitzades que, directa o indirectament han afectat en el desenvolupament del treball. Tot el programari que s'ha utliitzat durant el transcurs del treball és programari lliure, de codi obert i gratuït.

A.1 Sistemes operatius

Encara que la majoria de programes utilitzats tenen versions per a altres sistemes operatius, jo els he utilitzat sobre sistemes GNU/Linux, concretament en distribucions Ubuntu i Debian.

A.2 Processador de text LaTeX

Per tal de poder redactar el treball ha estat necessari un processador de textos especial. Ha calgut escriure un gran nombre de fórmules i expressions matemàtiques, pseudocodi, programes complets... i en aquest sentit LaTeX ha ajudat molt. LaTeX és un sistema de composició de text amb una "alta qualitat tipogràfica", on el text s'escriu en forma de codi i se li dona format amb ordres específiques incloses en el document. D'aquesta manera s'aconsegueix exactament el que es desitja, essent possible modificar molts dels paràmetres del document. Per exemple: la fórmula

$$e' = \frac{1}{2}\sum_{i=1}^{n} g(v_i)^2 - e$$

s'escriu a LaTeX com a

```
\[ e'=\frac{1}{2} \sum_{i=1}^{n}g(v_{i})^{2}-e \]
```

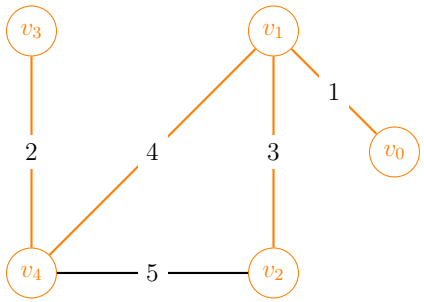
Posteriorment, tot el codi es compila i es genera un fitxer .pdf amb tot el codi interpretat.

Un altre avantatge de LaTeX és que també permet la utilització de diversos paquets que afegeixen funcionalitats addicionals. Durant la redacció d'aquesta memòria se n'han utilitzat un total de 36, i els següents són els més importants:

A.2.1 Tikz i tkz-berge

Tikz i tkz-berge permeten dibuixar (mitjançant ordres escrites) directament al document, sense que sigui necessari importar imatges externes. La gran majoria de imatges fetes per mi mateix s'han fet a través d'aquests dos paquets. Aquests, al mateix temps, tenen un sistema propi de paquets que permet afegir funcions i ordres. Dins d'aquests paquets, n'hi ha que inclouen diverses funcions orientades a grafs i permeten dibuixar-los tal com es desitgin.

La següent imatge d'un graf



s'escriu a LaTeX com a

```
\begin{figure}[H]
        \centering
        \begin{tikzpicture}[round/.style={circle, draw=black, thin,
 ↪  minimum size=2mm}, transform shape, scale=0.8]
                \node[round,orange] (3) at (0,4) {$v_{3}$};
                \node[round,orange] (4) at (0,0) {$v_{4}$};
                \node[round,orange] (2) at (4,0) {$v_{2}$};
                \node[round,orange] (1) at (4,4) {$v_{1}$};
                \node[round,orange] (0) at (6,2) {$v_{0}$};
                \Edge[label=$5$](2)(4)
                \tikzset{EdgeStyle/.style={orange}}
                \Edge[label=$1$](0)(1)
                \Edge[label=$3$](1)(2)
                \Edge[label=$4$](1)(4)
                \Edge[label=$2$](3)(4)
        \end{tikzpicture}
\end{figure}
```

A.2.2 Minted i Algorithm2e

Per poder incloure codi en el treball també ha sigut necessari utilitzar diversos paquets. Minted és el paquet que permet introduïr codi amb el seu format corresponent i ressaltant la sintaxi, i Algorithm2e és l'encarregat de donar format al pseudocodi. Aquests dos paquets permeten que el codi i pseudocodi quedin diferenciats de la resta del text i que, a més, es respecti la forma del programa.

A.3 GeoGebra

GeoGebra és un programa multiús, útil tant en el camp de la geometria, com en l'àlgebra o el càlcul. En aquest cas ha estat utilitzat per a generar les imatges dels punts de Fermat i alguna altra figura concreta com el polígon del problema de les càmeres de seguretat.

A.4 Python

Python és un lenguatge de propòsit general i d'alt nivell que busca senzillesa i eficiència, fent que la sintaxi sigui més planera. Aquest llenguatge permet escriure programes més entenedors i en menys línies que altres llenguatges com C/C++.

A.5 Git

Per tal de mantenir actualitzats els fitxer de redacció de la memòria i el fitxer amb els programes en Python he utilitzat Git. Git és un sistema de control de versions, és a dir, un conjunt d'eines que permeten tenir una còpia remota dels fitxers a la qual se li van afegint els canvis de cada sistema local. Per exemple: des d'un ordinador es realitza un canvi en el fitxer de la memòria del treball. Aquests canvis locals es sincronitzen amb la còpia remota i queda arxivat com a una nova versió. Quan es vol treballar des d'un altre ordinador es comprova si la versió local correspon a la versió remota, és a dir, es mira si hi ha hagut canvis que encara no té. En cas afirmatiu, es descarreguen els canvis de la còpia remota i s'integren a la còpia local. D'aquesta manera sempre es té la darrera versió i sempre hi ha un registre de tots els canvis que, a més, permet tornar a una versió anterior si fos necessari.

Annex B

Implementació d'algorismes

Per tal de representar els grafs i introduir-los als algorismes s'han utilitat llistes d'adjacència. L'ús de llistes d'adjacència es basa en un dicionari de diccionaris en cas de grafs ponderats, i un dicionari de llistes en cas de grafs no ponderats.

Per exemple: el graf (a) de la figura B.1 té com a llista d'adjacència

```
a={0:[1], 1:[3], 2:[0,3], 3:[]}
```

i la del graf (b) és

```
b={0:{1:4, 2:3}, 1:{2:-2}, 2:{}}
```

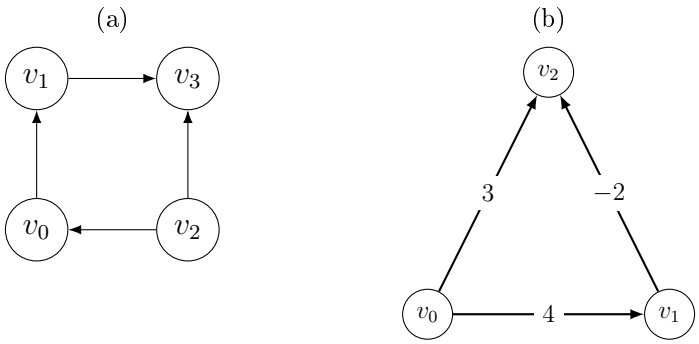


Figura B.1: Exemples de grafs

B.1 DFS

Codi

```
parent={}
topo=[]
def DFS(Adj):
    node=[]
    for i in range(0, len(Adj)):
        node.append(i)

    for s in node:
        if s not in parent:
            print "Des del node %d es pot arribar a:" %s
            print s
            parent[s]=None
            DFS_recursive(Adj, s)
    print "Ordre de recursió (Ordenació topològica per a grafs dirigits
 ↪  acíclics):"
    topo.reverse()
    print topo


def DFS_recursive(Adj, s):
    for v in Adj[s]:
        if v not in parent:
            print v
            parent[v]=s
            DFS_recursive(Adj, v)
    topo.append(s)
```

Exemple d'entrada

```
G={0:[1,2,3], 1:[3,4], 2:[4], 3:[4], 4:[]}
print DFS(G)
```

Exemple de sortida

```
Des de 0 es pot arribar a:
0
1
3
4
2
Ordre de recursió (Ordenació topològica per a grafs dirigits acíclics):
[0, 2, 1, 3, 4]
```

B.2 BFS

Codi

```
def BFS(Adj, s):
    level={s:0}
    parent={s:None}
    i=1
    frontier=[s]
    print s
    while frontier:
        next=[]
        for u in frontier:
            for v in Adj[u]:
                if v not in  level:
                    level[v]=i
                    parent[v]=u
                    next.append(v)
                    print v
        frontier=next
        i+=1
    print level
```

Exemple d'entrada

```
G={0:[1,2,3], 1:[3,4], 2:[4], 3:[4], 4:[]}
print BFS(G, 0)
```

Exemple de sortida

```
{0: 0, 1: 1, 2: 1, 3: 1, 4: 2}
```

B.3 Dijkstra

Codi

```
def Dijkstra(Adj, s):
    Q={}
    dist={}
    tree={}
    for i in range(0, len(Adj)):
        Q[i]=float("inf")
        dist[i]=float("inf")
    Q[s]=0
    while Q:
        u = min(Q, key=Q.get)
        dist[u] = Q[u]
        for v in Adj[u]:
            if v in Q:
                if Q[v] > Q[u] + Adj[u][v]:
                    Q[v] = Q[u] + Adj[u][v]
                    tree[v] = u
        Q.pop(u)

    return dist, tree

def OrderedDijkstra(Adj, s, k):
    Q = dict.fromkeys(Adj.keys(), float("inf"))
    dist = dict.fromkeys(Adj.keys(), float("inf"))
    tree = {}
    Q[s] = 0
    while Q:
        u = min(Q, key=Q.get)
        dist[u] = Q[u]
        for v in Adj[u]:
            if v in Q:
                if Q[v] > Q[u] + Adj[u][v]:
                    Q[v] = Q[u] + Adj[u][v] + k
                    tree[v] = u
        Q.pop(u)

    return dist, tree
```

Exemple d'entrada

```
G={0:{1:10,2:3}, 1:{2:1, 3:2}, 2:{1:4, 3:8, 4:2}, 3:{4:7}, 4:{3:9}}
distancies, arbre_anteriors = Dijkstra(G, 0)
print distancies
print arbre_anteriors
```

Exemple de sortida

```
{0: 0, 1: 7, 2: 3, 3: 9, 4: 5}
{1: 2, 2: 0, 3: 1, 4: 2}
```

B.4 Bellman-Ford

Codi

```
def BellmanFord(Adj, s):
    dist={}
    tree={}
    for i in range(0, len(Adj)):
        dist[i]=float("inf")
        tree[i]=None
    dist[s]=0

    for i in range(0, len(Adj)-1):
        for u in range(0, len(Adj)):
            for v in Adj[u]:
                if dist[v] > dist[u] + Adj[u][v]:
                    dist[v] = dist[u] + Adj[u][v]
                    tree[v]=u
    for u in range(0, len(Adj)):
        for v in Adj[u]:
            if dist[v] > dist[u] + Adj[u][v]:
                print "Hi ha cicles de pesos negatius"
                break
    return dist, tree
```

Exemple d'entrada

```
G={0:{1:16,2:0}, 1:{2:-32}, 2:{3:8,4:0}, 3:{4:-16}, 4:{5:4,6:0},
 ↪  5:{6:-8}, 6:{7:2,8:0}, 7:{8:-4}, 8:{9:1,10:0}, 9:{10:-2}, 10:{}}
distancies, arbre_apuntadors = BellmanFord(G, 0)
print distancies
print arbre_apuntadors
```

Exemple de sortida

```
{0: 0, 1: 16, 2: -16, 3: -8, 4: -24, 5: -20, 6: -28, 7: -26, 8: -30, 9:
 ↪  -29, 10: -31}
{0: None, 1: 0, 2: 1, 3: 2, 4: 3, 5: 4, 6: 5, 7: 6, 8: 7, 9: 8, 10: 9}
```

B.5 Prim

Codi

```
def Prim(Adj):
    Q={}
    tree={}
    for i in range(0,len(Adj)):
        Q[i]=float("inf")
    Q[0]=0
    while Q:
        u = min(Q, key=Q.get)
        for v in Adj[u]:
            if v in Q and Adj[u][v] < Q[v]:
                Q[v] = Adj[u][v]
                tree[v] = u
        Q.pop(u)
    return tree
```

Exemple d'entrada

```
G={1:{2:1, 3:2}, 0:{1:10,2:3}, 2:{1:4, 3:8, 4:2}, 4:{3:9}, 3:{4:7}}
print Prim(G)
```

Exemple de sortida

```
{1: 2, 2: 0, 3: 1, 4: 2}
```

B.6 Kruskal

Codi

```
from UnionFind import UnionFind #una bona implementació de UnionFind es
 ↪  troba a https://www.ics.uci.edu/~eppstein/PADS/UnionFind.py
def Kruskal(Adj):
    subtree = UnionFind()
    tree = []
    for e, u, v in sorted((Adj[u][v],u,v) for u in Adj for v in Adj[u]):
        for u in Adj:
            for v in Adj[u]:
                if subtree[u] != subtree[v]:
                    tree.append((u,v))
                    subtree.union(u,v)
    return tree
```

Exemple d'entrada

```
G={0:{1:8,3:2}, 1:{0:8,2:5,3:3,4:1}, 2:{1:5,3:2,4:4}, 3:{0:2,1:3,2:2},
 ↪  4:{1:1,2:4}}
print Prim(G)
```

Exemple de sortida

```
[(0, 1), (0, 3), (1, 2), (1, 4)]
```

B.7 Floyd-Warshall

Codi

```
def FloydWarshall(Adj):
    dist=[[float("inf") for x in range(len(Adj))] for y in
 ↪  range(len(Adj))]
    for i in range(0,len(Adj)):
       dist[i][i] = 0
    for v in range(len(Adj)):
        for u in Adj[v]:
            dist[v][u] = Adj[v][u]
    for x in range(len(Adj)):
        for u in range(len(Adj)):
            for v in range(len(Adj)):
                if dist[u][v] > dist[u][x] + dist[x][v]:
                    dist[u][v] = dist[u][x] + dist[x][v]
    return dist
```

Exemple d'entrada

```
G={0:{1:10,2:3}, 1:{2:1, 3:2}, 2:{1:4, 3:8, 4:2}, 3:{4:7}, 4:{3:9}}
print FloydWarshall(G)
```

Exemple de sortida

```
[[0, 7, 3, 9, 5],
[inf, 0, 1, 2, 3],
[inf, 4, 0, 6, 2],
[inf, inf, inf, 0, 7],
[inf, inf, inf, 9, 0]]
```

B.8 Hamilton

Codi

```
def Hamilton_recursive(Adj, s, e, path):
    path = path + [s]
    if s == e:
        return path
    for n in Adj[s]:
        if n not in path:
            nou_path = Hamilton_recursive(Adj, n, e, path)
            if nou_path:
                return nou_path
    return None

def Hamilton(Adj, s, e):
    path=[]
    return Hamilton_recursive(Adj, s, e, path)
```

Exemple d'entrada

```
G={0:[2,3,5,1], 1:[0,2,4,5], 2:[0,1,3,4], 3:[0,2,4,5], 4:[1,2,3,5],
 ↪  5:[0,1,3,4]}
print Hamilton(G, 0, 5)
```

Exemple de sortida

```
[0, 2, 1, 4, 3, 5]
```

B.9 Euler

Codi

```
def Euler(Adj):
    graf = Adj
    senar = [v for v in graf.keys() if len(graf[v])%2 != 0]
    senar.append(graf.keys()[0])
    print senar

    if len(senar)>3:
        return None

    Q = [senar[0]]
    path = []
    while Q:
        v = Q[-1]
        if graf[v]:
            u = graf[v][0]
            Q.append(u)
            del graf[u][graf[u].index(v)]
            del graf[v][0]
        else:
            path.append(Q.pop())

    return path
```

Exemple d'entrada

```
G={0:[1,2,3],1:[0,2,3],2:[0,1,3,4],3:[0,1,2,4],4:[3,2]}
print Euler(G)
```

Exemple de sortida

```
[1, 3, 4, 2, 3, 0, 2, 1, 0]
```

B.10 Coloració

Codi

```
def coloring(Adj):
    graph = sorted(Adj, key=lambda k:len(Adj[k]), reverse=True)
    colors = {}
    usat = False
    actual = 0

    for i in range(0, len(Adj)):
        colors[i]=None
    colors[graph[0]]=0

    while None in colors.values():
        for v in graph:
            if colors[v] == None:
                for k in Adj[v]:
                    if colors[k] == actual:
                        usat = True
                        break

                if usat == False:
                    colors[v] = actual
                usat = False
        actual = actual + 1
    return colors
```

Exemple d'entrada

```
bipartite={0:[4,5,6,7], 1:[4,5,6,7], 2:[4,5,6,7], 3:[4,5,6,7],
 ↪  4:[0,1,2,3], 5:[0,1,2,3], 6:[0,1,2,3], 7:[0,1,2,3]}
print coloring(bipartite)
```

Exemple de sortida

```
{0: 0, 1: 0, 2: 0, 3: 0, 4: 1, 5: 1, 6: 1, 7: 1}
```

B.11 Metro

Codi

```
def metro(Adj, inici, final, k): #on k és el temps de parada acada
↪  estació
    recorregut=[]

    print "Punt inicial:", inici.decode("ISO-8859-15")

    print "Punt final:", final.decode("ISO-8859-15")

    dist, tree = OrderedDijkstra(Adj, inici, k)

    i = final
    while tree[i] != inici:
        recorregut.append(tree[i])
        i = tree[i]

    recorregut.append(inici)
    recorregut.reverse()

    total= dist[final]-k

    print "Temps amb estacions del recorregut:", dist[final], "Temps
↪  real:", total

    if total < 60:
        print "Temps total del recorregut:",int(total), "segons"
    else:
        minuts = total/60
        segons = (total%60)
        print "Temps total del recorregut:", int(minuts),"minuts i",
↪  int(segons), "segons"

    print "Recorregut:",
    print "[",
    for i in range(0,len(recorregut)):
        print recorregut[i].decode("ISO-8859-15")+",",

    print final.decode("ISO-8859-15"),"]"
```

Exemple d'entrada

```
metro_barcelona={"1_Hospital de Bellvitge":{"1_Bellvitge":90},
↪  "1_Bellvitge":{"1_Hospital de Bellvitge":90, "1_Av.
↪  Carrilet":100},"1_Av. Carrilet":{"1_Bellvitge":100, "1_Rbla. Just
↪  Oliveras":65, "8_L'Hospitalet - Av. Carrilet":180}, "1_Rbla. Just
↪  Oliveras":{"1_Av. Carrilet":65, "1_Can Serra":60}, "1_Can
↪  Serra":{"1_Rbla. Just Oliveras":60, "1_Florida":60},
↪  "1_Florida":{"1_Can Serra":60, "1_Torrassa":60},
↪  "1_Torrassa":{"1_Florida":60, "1_Santa Eulàlia":85,
↪  "9S_Torrassa":240}, "1_Santa Eulàlia":{"1_Torrassa":85,"1_Mercat
↪  Nou":75}, "1_Mercat Nou":{"1_Santa Eulàlia":75, "1_Plaça de
↪  Sants":60}, "1_Plaça de Sants":{"1_Mercat Nou":60,
↪  "1_Hostafrancs":50, "5_Plaça de Sants":282},
↪  "1_Hostafrancs":{"1_Plaça de Sants:":50, "1_Espanya":55},
↪  "1_Espanya":{"1_Hostafrancs":55, "1_Rocafort":60, "3_Espanya":209,
↪  "8_Espanya":120}, "1_Rocafort":{"1_Espanya":60, "1_Urgell":55},
↪  "1_Urgell":{"1_Rocafort":55, "1_Universitat":58},
↪  "1_Universitat":{"1_Urgell":58, "1_Catalunya":50,
↪  "2_Universitat":144}, "1_Catalunya":{"1_Universitat":50,
↪  "1_Urquinaona":58, "3_Catalunya":180, "6_Catalunya":360,
↪  "7_Catalunya":360}, "1_Urquinaona":{"1_Catalunya":58, "1_Arc de
↪  Triomf":85, "4_Urquinaona":256}, "1_Arc de
↪  Triomf":{"1_Urquinaona":85, "1_Marina":54}, "1_Marina":{"1_Arc de
↪  Triomf":54, "1_Glòries":80}, "1_Glòries":{"1_Marina":80,
↪  "1_Clot":78}, "1_Clot":{"1_Glòries":78,"1_Navas":65, "2_Clot":120},
↪  "1_Navas":{"1_Clot":65, "1_La Sagrera":80}, "1_La
↪  Sagrera":{"1_Navas":80, "1_Fabra i Puig":89, "5_La Sagrera":100,
↪  "9N_La Sagrera":178, "10_La Sagrera":178}, "1_Fabra i Puig":{"1_La
↪  Sagrera":89, "1_Sant Andreu":101}, "1_Sant Andreu":{"1_Fabra i
↪  Puig":101, "1_Torras i Bages":88}, "1_Torras i Bages":{"1_Sant
↪  Andreu":88, "1_Trinitat Vella":77}, "1_Trinitat Vella":{"1_Torras i
↪  Bages":77, "1_Baró de Viver":60}, "1_Baró de Viver":{"1_Trinitat
↪  Vella":60, "1_Santa Coloma":83}, "1_Santa Coloma":{"1_Baró de
↪  Viver":83, "1_Fondo":84}, "1_Fondo":{"1_Santa Coloma":84,
↪  "9N_Fondo":140}, "2_Paral·lel":{"2_Sant Antoni":80, "3_Paral·lel":83},
↪  "2_Sant Antoni":{"2_Paral·lel":80, "2_Universitat":66},
↪  "2_Universitat":{"2_Sant Antoni":66, "2_Passeig de Gràcia":80,
↪  "1_Universitat":144}, "2_Passeig de Gràcia":{"2_Universitat":80,
↪  "2_Tetuan":95, "3_Passeig de Gràcia":360, "4_Passeig de Gràcia":120},
↪  "2_Tetuan":{"2_Passeig de Gràcia":95, "2_Monumental":93},
↪  "2_Monumental":{"2_Tetuan":93, "2_Sagrada Família":63}, "2_Sagrada
↪  Família":{"2_Monumental":63, "2_Encants":114, "5_Sagrada
↪  Família":178}, "2_Encants":{"2_Sagrada Família":114, "2_Clot":53},
↪  "2_Clot":{"2_Encants":53, "2_Bac de Roda":89, "1_Clot":120}, "2_Bac
↪  de Roda":{"2_Clot":89, "2_Sant Martí":61}, "2_Sant Martí":{"2_Bac de
↪  Roda":61, "2_La Pau":74}, "2_La Pau":{"2_Sant Martí":74,
↪  "2_Verneda":76, "4_La Pau":60}, "2_Verneda":{"2_La Pau":76,
↪  "2_Artigues Sant Adrià":81}, "2_Artigues Sant Adrià":{"2_Verneda":81,
↪  "2_Sant Roc":91}, "2_Sant Roc":{"2_Artigues Sant Adrià":91,
↪  "2_Gorg":60},
```

```
"2_Gorg":{"2_Sant Roc":60, "2_Pep Ventura":58, "10_Gorg":78}, "2_Pep
↪  Ventura":{"2_Gorg":58, "2_Badalona Pompeu Fabra":74}, "2_Badalona
↪  Pompeu Fabra":{"2_Pep Ventura":74}, "3_Zona Universitària":{"3_Palau
↪  Reial":67, "9S_Zona Universitària":265}, "3_Palau Reial":{"3_Zona
↪  Universitària":67, "3_Maria Cristina":65}, "3_Maria
↪  Cristina":{"3_Palau Reial":65, "3_Les Corts":68}, "3_Les
↪  Corts":{"3_Maria Cristina":68, "3_Plaça del Centre":63}, "3_Plaça del
↪  Centre":{"3_Les Corts":63, "3_Sants Estació":57},"3_Sants
↪  Estació":{"3_Plaça del Centre":57, "3_Tarragona":55, "5_Sants
↪  Estació":232},"3_Tarragona":{"3_Sants Estació":55, "3_Espanya":65},
↪  "3_Espanya":{"3_Tarragona":65, "3_Poble Sec":67, "1_Espanya":209,
↪  "8_Espanya":360}, "3_Poble Sec":{"3_Espanya":67, "3_Paral·lel":70},
↪  "3_Paral·lel":{"3_Poble Sec":70, "3_Drassanes":72, "2_Paral·lel":83},
↪  "3_Drassanes":{"3_Paral·lel":72, "3_Liceu":75},
↪  "3_Liceu":{"3_Drassanes":75, "3_Catalunya":60},
↪  "3_Catalunya":{"3_Liceu":60, "3_Passeig de Gràcia":70,
↪  "1_Catalunya":180, "6_Catalunya":180, "7_Catalunya":180}, "3_Passeig
↪  de Gràcia":{"3_Catalunya":70, "3_Diagonal":77, "2_Passig de
↪  Gràcia":360, "4_Passeig de Gràcia":238}, "3_Diagonal":{"3_Passeig de
↪  Gràcia":77, "3_Fontana":75, "5_Diagonal":217, "6_Provença":360,
↪  "7_Provença":360}, "3_Fontana":{"3_Diagonal":75, "3_Lesseps":55},
↪  "3_Lesseps":{"3_Fontana":55, "3_Vallcarca":85},
↪  "3_Vallcarca":{"3_Lesseps":85, "3_Penitents":90},
↪  "3_Penitents":{"3_Vallcarca":90, "3_Vall d'Hebron":80}, "3_Vall
↪  d'Hebron":{"3_Penitents":80, "3_Montbau":62, "5_Vall d'Hebron":204},
↪  "3_Montbau":{"3_Vall d'Hebron":62, "3_Mundet":64},
↪  "3_Mundet":{"3_Montbau":64, "3_Valldaura":78},
↪  "3_Valldaura":{"3_Mundet":78, "3_Canyelles":79},
↪  "3_Canyelles":{"3_Valldaura":79, "3_Roquetes":94},
↪  "3_Roquetes":{"3_Canyelles":94, "3_Trinitat Nova":84}, "3_Trinitat
↪  Nova":{"3_Roquetes":84, "4_Trinitat Nova":210, "11_Trinitat
↪  Nova":210}, "4_Trinitat Nova":{"4_Via Júlia":99, "3_Trinitat
↪  Nova":210, "11_Trinitat Nova":0}, "4_Via Júlia":{"4_Trinitat
↪  Nova":99, "4_Llucmajor":87}, "4_Llucmajor":{"4_Via Júlia":87,
↪  "4_Maragall":161}, "4_Maragall":{"4_Llucmajor":161, "4_Guinardó
↪  Hospital de Sant Pau":88, "5_Maragall":191}, "4_Guinardó Hospital de
↪  Sant Pau":{"4_Maragall":88, "4_Alfons X":86}, "4_Alfons
↪  X":{"4_Guinardó Hospital de Sant Pau":86, "4_Joanic":77},
↪  "4_Joanic":{"4_Alfons X":77, "4_Verdaguer":89},
↪  "4_Verdaguer":{"4_Joanic":89, "4_Girona":86, "5_Verdaguer":218},
↪  "4_Girona":{"4_Verdaguer":86, "4_Passeig de Gràcia":83}, "4_Passeig
↪  de Gràcia":{"4_Girona":83, "4_Urquinaona":92, "2_Passeig de
↪  Gràcia":120, "3_Passeig de Gràcia":238}, "4_Urquinaona":{"4_Passeig
↪  de Gràcia":92, "4_Jaume I":60, "1_Urquinaona":256}, "4_Jaume
↪  I":{"4_Urquinaona":60, "4_Barceloneta":78}, "4_Barceloneta":{"4_Jaume
↪  I":78, "4_Ciutadella Vila Olímpica":82}, "4_Ciutadella Vila
↪  Olímpica":{"4_Barceloneta":82, "4_Bogatell":113},
↪  "4_Bogatell":{"4_Ciutadella Vila Olímpica":113, "4_Llacuna":62},
↪  "4_Llacuna":{"4_Bogatell":62, "4_Poblenou":64},
↪  "4_Poblenou":{"4_Llacuna":64, "4_Selva del Mar":66}, "4_Selva del
↪  Mar":{"4_Poblenou":66, "4_El Maresme Fòrum":86}, "4_El Maresme
↪  Fòrum":{"4_Selva del Mar":86, "4_Besòs Mar":65},
```

```
"4_Besòs Mar":{"4_El Maresme Fòrum":65, "4_Besòs":67},
↪  "4_Besòs":{"4_Besòs Mar":67, "4_La Pau":76}, "4_La
↪  Pau":{"4_Besòs":76, "2_La Pau":60}, "5_Cornellà
↪  Centre":{"5_Gavarra":90}, "5_Gavarra":{"5_Cornellà Centre":90,
↪  "5_Sant Ildefons":85}, "5_Sant Ildefons":{"5_Gavarra":85, "5_Can
↪  Boixeres":74}, "5_Can Boixeres":{"5_Sant Ildefons":74, "5_Can
↪  Vidalet":90}, "5_Can Vidalet":{"5_Can Boixeres":90, "5_Pubilla
↪  Cases":83}, "5_Pubilla Cases":{"5_Can Vidalet":83,
↪  "5_Collblanc":105}, "5_Collblanc":{"5_Pubilla Cases":105,
↪  "5_Badal":70, "9S_Collblanc":240}, "5_Badal":{"5_Collblanc":70,
↪  "5_Plaça de Sants":74}, "5_Plaça de Sants":{"5_Badal":74, "5_Sants
↪  Estació":79, "1_Plaça de Sants":282}, "5_Sants Estació":{"5_Plaça de
↪  Sants":79, "5_Entença":73, "3_Sants Estació":232},
↪  "5_Entença":{"5_Sants Estació":73, "5_Hospital Clínic":64},
↪  "5_Hospital Clínic":{"5_Entença":64, "5_Diagonal":78},
↪  "5_Diagonal":{"5_Hospital Clínic":78, "5_Verdaguer":78,
↪  "3_Diagonal":217, "6_Provença":180, "7_Provença":180},
↪  "5_Verdaguer":{"5_Diagonal":78, "5_Sagrada Família":75,
↪  "4_Verdaguer":218}, "5_Sagrada Família":{"5_Verdaguer":75, "5_Sant
↪  Pau Dos de Maig":85, "2_Sagrada Família":178}, "5_Sant Pau Dos de
↪  Maig":{"5_Sagrada Família":85, "5_Camp de l'Arpa":63}, "5_Camp de
↪  l'Arpa":{"5_Sant Pau Dos de Maig":63, "5_La Sagrera":93}, "5_La
↪  Sagrera":{"5_Camp de l'Arpa":93, "5_Congrés":85, "1_La Sagrera":100,
↪  "9N_La Sagrera":233, "10_La Sagrera":233}, "5_Congrés":{"5_La
↪  Sagrera":85, "5_Maragall":60}, "5_Maragall":{"5_Congrés":60,
↪  "5_Virrei Amat":60, "4_Maragall":191}, "5_Virrei
↪  Amat":{"5_Maragall":60, "5_Vilapicina":74}, "5_Vilapicina":{"5_Virrei
↪  Amat":74, "5_Horta":75}, "5_Horta":{"5_Vilapicina":75, "5_El
↪  Carmel":78}, "5_El Carmel":{"5_Horta":78, "5_El Coll La
↪  Teixonera":80}, "5_El Coll La Teixonera":{"5_El Carmel":80, "5_Vall
↪  d'Hebron":81}, "5_Vall d'Hebron":{"5_El Coll La Teixonera":81,
↪  "3_Vall d'Hebron":204}, "6_Catalunya":{"6_Provença":120,
↪  "1_Catalunya":360, "3_Catalunya":180, "7_Catalunya":5},
↪  "6_Provença":{"6_Catalunya":120, "6_Gràcia":120, "3_Diagonal":360,
↪  "5_Diagonal":180, "7_Provença":5}, "6_Gràcia":{"6_Provença":120,
↪  "6_Sant Gervasi":60, "7_Gràcia":5}, "6_Sant Gervasi":{"6_Gràcia":60,
↪  "6_Muntaner":120, "7_Plaça Molina":5}, "6_Muntaner":{"6_Sant
↪  Gervasi":120, "6_La Bonanova":60}, "6_La Bonanova":{"6_Muntaner":60,
↪  "6_Les Tres Torres":120}, "6_Les Tres Torres":{"6_La Bonanova":120,
↪  "6_Sarrià":60}, "6_Sarrià":{"6_Les Tres Torres":60, "6_Reina
↪  Elisenda":120}, "6_Reina
↪  Elisenda":{"6_Sarrià":120},"7_Catalunya":{"7_Provença":120,
↪  "1_Catalunya":360,"3_Catalunya":180,
↪  "6_Catalunya":5},"7_Provença":{"7_Catalunya":120, "7_Gràcia":120,
↪  "3_Diagonal":360, "5_Diagonal":180,
↪  "6_Provença":5},"7_Gràcia":{"7_Provença":120,"7_Plaça
↪  Molina":60,"6_Gràcia":5},"7_Plaça
↪  Molina":{"7_Gràcia":60,"7_Pàdua":120, "6_Sant
↪  Gervasi":5},"7_Pàdua":{"7_Plaça Molina":120,"7_El Putxet":60},"7_El
↪  Putxet":{"7_Pàdua":60, "7_Av. Tibidabo":60}, "7_Av. Tibidabo":{"7_El
↪  Putxet":60}, "8_Espanya":{"8_Magòria La Campana":120,
↪  "1_Espanya":120, "3_Espanya":360}, "8_Magòria La
↪  Campana":{"8_Espanya":120, "8_Ildefons Cerdà":120},
```

```
"8_Ildefons Cerdà":{"8_Magòria La Campana":120, "8_Europa Fira":120},
 ↪  "8_Europa Fira":{"8_Ildefons Cerdà":120, "8_Gornal":120, "9S_Europa
 ↪  Fira":180}, "8_Gornal":{"8_Europa Fira":120, "8_Sant Josep":120},
 ↪  "8_Sant Josep":{"8_Gornal":120, "8_L'Hospitalet - Av. Carrilet":60},
 ↪  "8_L'Hospitalet - Av. Carrilet":{"8_Sant Josep":60, "8_Almeda":180,
 ↪  "1_Av. Carrilet":180}, "8_Almeda":{"8_L'Hospitalet - Av.
 ↪  Carrilet":180, "8_Cornellà - Riera":120}, "8_Cornellà -
 ↪  Riera":{"8_Almeda":120, "8_Sant Boi":180}, "8_Sant Boi":{"8_Cornellà
 ↪  - Riera":180, "8_Molí Nou - Ciutat Cooperativa":120}, "8_Molí Nou -
 ↪  Ciutat Cooperativa":{"8_Sant Boi":120}, "9N_La Sagrera":{"9N_Onze de
 ↪  Setembre":116, "1_La Sagrera":178, "5_La Sagrera":233, "10_La
 ↪  Sagrera":5}, "9N_Onze de Setembre":{"9N_La Sagrera":116, "9N_Bon
 ↪  Pastor":115, "10_Onze de Setembre":5}, "9N_Bon Pastor":{"9N_Onze de
 ↪  Setembre":114, "9N_Can Peixauet":125, "10_Bon Pastor":5}, "9N_Can
 ↪  Peixauet":{"9N_Bon Pastor":125, "9N_Santa Rosa":65}, "9N_Santa
 ↪  Rosa":{"9N_Can Peixauet":65, "9N_Fondo":87}, "9N_Fondo":{"9N_Santa
 ↪  Rosa":87, "9N_Església Major":72, "1_Fondo":140}, "9N_Església
 ↪  Major":{"9N_Fondo":72, "9N_Singuerlín":76},
 ↪  "9N_Singuerlín":{"9N_Església Major":76, "9N_Can Zam":103}, "9N_Can
 ↪  Zam":{"9N_Singuerlín":103}, "9S_Aeroport T1":{"9S_Aeroport T2":240},
 ↪  "9S_Aeroport T2":{"9S_Aeroport T1":240, "9S_Mas Blau":80}, "9S_Mas
 ↪  Blau":{"9S_Aeroport T2":80, "9S_Parc Nou":120}, "9S_Parc
 ↪  Nou":{"9S_Mas Blau":120, "9S_Cèntric":85}, "9S_Cèntric":{"9S_Parc
 ↪  Nou":85, "9S_El Prat Estació":95}, "9S_El Prat
 ↪  Estació":{"9S_Cèntric":95, "9S_Les Moreres":150}, "9S_Les
 ↪  Moreres":{"9S_El Prat Estació":150, "9S_Mercabarna":100},
 ↪  "9S_Mercabarna":{"9S_Les Moreres":100, "9S_Parc Logístic":120},
 ↪  "9S_Parc Logístic":{"9S_Mercabarna":120, "9S_Fira":125},
 ↪  "9S_Fira":{"9S_Parc Logístic":125, "9S_Europa Fira":70}, "9S_Europa
 ↪  Fira":{"9S_Fira":70, "9S_Can Tries Gornal":70, "8_Europa Fira":180},
 ↪  "9S_Can Tries Gornal":{"9S_Europa Fira":70, "9S_Torrassa":75},
 ↪  "9S_Torrassa":{"9S_Can Tries Gornal":75, "9S_Collblanc":110,
 ↪  "1_Torrassa":240}, "9S_Collblanc":{"9S_Torrassa":110, "9S_Zona
 ↪  Universitària":105, "5_Collblanc":240}, "9S_Zona
 ↪  Universitària":{"9S_Collblanc":105, "3_Zona Universitària":265},
 ↪  "10_La Sagrera":{"10_Onze de Setembre":116, "1_La Sagrera":178, "5_La
 ↪  Sagrera":233, "9N_La Sagrera":5}, "10_Onze de Setembre":{"10_La
 ↪  Sagrera":116, "10_Bon Pastor":115, "9N_Onze de Setembre":5}, "10_Bon
 ↪  Pastor":{"10_Onze de Setembre":115, "10_Llefià":133, "9N_Bon
 ↪  Pastor":5}, "10_Llefià":{"10_Bon Pastor":133, "10_La Salut":59},
 ↪  "10_La Salut":{"10_Llefià":59, "10_Gorg":111}, "10_Gorg":{"10_La
 ↪  Salut":111, "2_Gorg":78}, "11_Trinitat Nova":{"11_Casa de
 ↪  l'Aigua":43, "3_Trinitat Nova":210, "4_Trinitat Nova":0}, "11_Casa de
 ↪  l'Aigua":{"11_Trinitat Nova":43, "11_Torre Baró Vallbona":111},
 ↪  "11_Torre Baró Vallbona":{"11_Casa de l'Aigua":111, "11_Ciutat
 ↪  Meridiana":60}, "11_Ciutat Meridiana":{"11_Torre Baró Vallbona":60,
 ↪  "11_Can Cuiàs":45}, "11_Can Cuiàs":{"11_Ciutat Meridiana":45}}

metro(metro_barcelona, "2_Paral·lel", "11_Casa de l'Aigua", 25)
```

Exemple de sortida

```
Punt inicial: 2_Paral·lel
Punt final: 11_Casa de l'Aigua
Temps net del recorregut: 1595
Temps total del recorregut: 26 minuts i 35 segons
Recorregut: [ 2_Paral·lel, 2_Sant Antoni, 2_Universitat, 2_Passeig de
 ↪  Gràcia, 4_Passeig de Gràcia, 4_Girona, 4_Verdaguer, 4_Joanic,
 ↪  4_Alfons X, 4_Guinardó Hospital de Sant Pau, 4_Maragall, 4_Llucmajor,
 ↪  4_Via Júlia, 4_Trinitat Nova, 11_Trinitat Nova, 11_Casa de l'Aigua ]
```