Sofie Jensen er netop færdig i gymnasiet, og har bestemt sig for (til sine forældres gru) at skyde hele sin børneopsparing af på en rejse. Selvom Sofies forældre har sparet op siden Sofies dåb er Sofie ikke helt tilfreds med størrelsen på opsparingen og efter nøje at have tjekket priser på flybilletter til henholdsvis Østasien og Mellemamerika har hun måtte sande, at opsparingen ikke slår til. Hun bestemmer sig derfor for, at hendes rejse skal gå rundt i Europa i stedet og så skal den foregå med tog! Sofie er sikker på, at hendes veninden Greta vil synes om hendes valg af transportmiddel.

Sofie har 14 dages ferie efter den sidste studenterfest før hun skal begynde på sit nye job og det er derfor vigtigt for Sofie, at hun får udnytte disse 14 dage så effektivt som muligt. Planen er derfor at bruge præcis 1 dag i en række europæiske hovedstæder, hvor den så eller står på sightseeing *"extravaganza"*.

Hun er netop kommet til planlægningen af sin dag i Paris. Hun har Googlet en del og har lavet en list over de 10 vigtigste seværdigheder som består af følgende

10 vigtigste seværdigheder i Paris	
Notre Dame	Eiffeltårnet
Rue des Martyrs	Sacré Coeur
Palais du Luxembourg	Louvre
Triumfbuen	Moulin Rouge
Pont de Bir-Hakeim	Latiner Kvarteret

Sofie ankommer med nattoget til Gare du Nord klokken 6.30 fra Madrid. Det første hun vil gøre er at gå til sit hotel for at tjekke ind og for at have et sted at efterlade sin bagage. Hun regner med, at det alt i alt tager hende halvanden time så hun er klar til at forlade hotellet klokken 8.00.

Sofie har været på Google Maps og har fundet afstandene mellem alle seværdighederne samt sit hotel og i samme ombæring har hun brugt den funktion som giver hende et estimat over hvor lang tid det tager at gå denne distance. Disse data er givet i den vedlagte Excel-fil. Derudover han hun på forskellige fora på nettet spurgt andre rejsende om hvor længe hun skal regne med, at det tager at besøge de enkelte seværdigheder. Disse estimater er også vedlagt i Excel-filen.

Baseret på data fra Excel-filen ønsker Sofie hjælp til at planlægge sin dag i Paris.

1. Formuler en lineær heltals-model, som udregner hvor lang tid det tager at besøge de 10 seværdigheder, hvis den rejsende minimerer sit samlede tidforbrug på sightseeing-turen.
   Hint: for at inkludere tiden der bliver brugt på at se seværdighederne, kan rejsetidsmatricen ændres til $c_{ij} = rejsetid\ fra\ i\ til\ j + tiden\ brugt\ ved\ seværdighed\ i.$

   Besvarelse:
   1.1: For at kunne besvare denne opgave, starter vi med at definere en række parametre som skal bruges i modellen.

$$n = antal\ seværdigheder$$
$$I = \{1, \dots, n\} = mængde\ af\ alle\ seværdigheder$$
$$I_0 = \{0,1, \dots, n\} = mængde\ af\ alle\ lokationer, altså\ seværdigheder + hotel$$
$$t_{ij} = tid\ \text{det}\ tager\ at\ gå\ fra\ lokation\ i\ til\ lokation\ j\ i\ minutter$$
$$d_{ij} = afstand\ mellem\ lokation\ i\ og\ lokation\ j\ i\ kilometer$$
$$\tau_i = tiden\ det\ tager\ at\ besøge\ \ lokation\ i$$

   Her lader vi $\{0\}$ angive hotellet.

1.2: Med definition af disse parametre kan vi nu definere de variabler der skal være i spil i modellen:
For alle $i, j \in I_0$ defineres de binære variabler:

$$x_{ij} = \begin{cases} 1, hvis\ Sofie\ rejser\ direkte\ fra\ lokation\ i\ til\ lokation\ j \\ 0, ellers \end{cases}$$

For alle $i \in I$ defineres de kontinuerte variabler:

$$u_i = antallet\ af\ seværdigheder, som\ Sofie\ har\ besøgt, \\ når\ hun\ forlader\ seværdighed\ i$$

1.3: Vi kan nu opstille objektfunktionen:

$$\min \sum_{i \in I_0} \sum_{j \in I_0} (t_{ij} + \tau_i) x_{ij}$$

Vi minimerer altså den samlede rejsetid, plus den tid det tager at besøge seværdighederne.
1.4: Vi skal her opstille begrænsningerne for at opnå en TSP tur gennem alle seværdighederne. Den første begrænsning sørger for, at Sofie forlader hotellet præcis en gang

$$\sum_{j \in I} x_{0j} = 1$$

Og den næste sørger for, at hun kommer tilbage præcis en gang:

$$\sum_{i \in I} x_{i0} = 1$$

Det samme skal gøre sig gældende for alle seværdighederne, så vi får at hver seværdighed besøges en gang

$$\sum_{i \in I_0} x_{ij} = 1, \quad \forall j \in I$$

Og at alle seværdighederne forlades præcis en gang hver

$$\sum_{j \in I} x_{ij} = 1, \quad \forall i \in I$$

Slutteligt, skal vi tilføje de såkaldte "subtour elimination constraints" som her har følgende form

$$u_i - u_j + n x_{ij} = n - 1, \quad \forall i, j \in I$$

Det vil sige, at Sofies blandede heltalsmodel kommer til at se ud som følger:

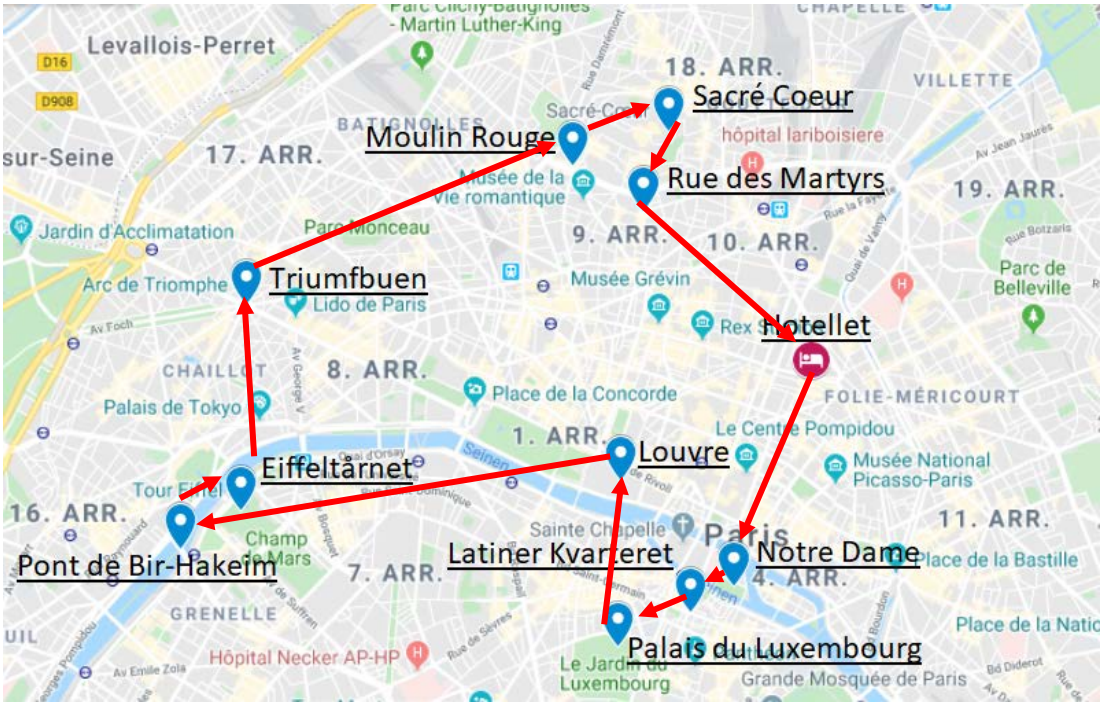
$$\begin{aligned} \min & \sum_{i \in I_0} \sum_{j \in I_0} (t_{ij} + \tau_i) x_{ij} \\ s.t.: & \sum_{j \in I} x_{0j} = 1 \\ & \sum_{i \in I} x_{i0} = 1 \\ & \sum_{i \in I_0} x_{ij} = 1, \quad \forall j \in I \\ & \sum_{j \in I} x_{ij} = 1, \quad \forall i \in I \\ & u_i - u_j + n x_{ij} \leq n - 1, \quad \forall i, j \in I \\ & x_{ij} \in \{0,1\}, \quad \forall i, j \in I_0 \\ & 0 \leq u_i \leq n, \quad \forall i \in I \end{aligned}$$

2. Implementer modellen i OPL og løs den vha. CPLEX. Illustrer løsningen på kortet (vedlagt i Excel-filen) og kommenter på dens karakteristika.

Besvarelse:
Modellen kan implementeres i OPL og ende med at se ud som følger:

```
int n = ...;
range I = 1..n;
range I_0 = 0..n;

int t[I_0][I_0] = ...;
float d[I_0][I_0] = ...;
int tau[I_0] =...;

dvar boolean x[I_0][I_0];
dvar float+ u[I] in I;


dexpr float TotalTime = sum ( i, j in I_0 ) ( tau[i] + t[i][j] ) * x[i][j];
dexpr float TotalDistance = sum ( i, j in I_0 ) d[i][j] * x[i][j];
dexpr float TotalTravelTime = sum ( i, j in I_0 ) t[i][j] * x[i][j];

minimize TotalTime;
subject to
{
    sum ( j in I ) x[0][j] == 1;
    sum ( i in I ) x[i][0] == 1;
    forall ( i in I ) sum ( j in I_0 ) x[i][j] == 1;
    forall ( j in I ) sum ( i in I_0 ) x[i][j] == 1;




    forall ( i,j in I )
     {
        u[i] - u[j] + n * x[i][j] <= n - 1;
     }
}
```

Løses modellen, fås en løsning givet ved



I alt skal Sofie være på sightseeing i 1.165 minutter, eller 19 timer og 25 minutter, hvilket vil sige, at hun kommer tilbage til sit hotel klokken 3.25 om natten, hvilket må siges at være lidt halv-sent. Hun kommer til at tilbagelægge 19.4 kilometer og i alt gå mellem seværdigheder i 250 minutter, eller 4 timer og 10 minutter.

Sofie kan se på løsningen til Opgave 2, at det kommer til at tage alt for lang tid at besøge alle de 10 seværdigheder. Hun er først tilbage på hotellet midt om natten og hun skal rejse videre allerede næste morgen, så det er desværre ikke en brugbar løsning for hende.

3. For at imødekomme Sofies bekymring bedes du nu formulere en ny model (eller ændre din eksisterende) således at antallet af seværdigheder, som Sofie kan nå at se *inden for 12 timer,* bliver maksimeret. *Hint: Introducer binære variable y_i for hver seværdighed som er lig med 1 hvis og kun hvis seværdighed i besøges og inkorporer dem i den eksisterende model.*

Besvarelse:

Vi indfører altså nu en binær variabel y_i for hver seværdighed, $i \in I$, som afgør om vi besøger seværdighed i eller ej. Matematisk defineres y_i som følger:

$$y_i = \begin{cases} 1, hvis\ seværdighed\ i \in I\ besøges\ på\ Sofies\ tur \\ 0, ellers \end{cases}$$

Vi skal nu ændre vores begrænsninger og objektfunktion fra før. Da vi bliver bedt om at maksimere antallet af seværdigheder, Sofie kan nå at se, skal vi blot skifte objektfunktionen fra opgave 1 ud med

$$\max \sum_{i \in I} y_i$$

Sofie skal stadig forlade hotellet og ankomme til hotellet præcis en gang, så disse begrænsninger skal der ikke ændres på. Vi skal dog ikke nødvendigvis ankomme og forlade hver seværdighed mere. Det er kun hvis vi rent faktisk vælger at besøge seværdigheden. Det vil sige, vi kan ændre disse begrænsninger til

$$\sum_{i \in I_0} x_{ij} = y_j, \qquad \forall j \in I$$

$$\sum_{j \in I_0} x_{ij} = y_i, \quad \forall i \in I$$

Vi skal også sørge for, at den samlede tid Sofie bruger på sightseeing ikke overstiger 12 timer, hvilket vil sige, at vi blot skal begrænse den gamle objektfunktion, som målte den totale tid brugt på sightseeing, med 12 timer:

$$\sum_{i \in I_0} \sum_{j \in I_0} (t_{ij} + \tau_i) x_{ij} \leq 12 \cdot 60 = 720$$

Vi husker lige, at vi måler alting i minutter hvorfor højre-siden skal ganges med 60 minutter.
Sub tour eliminerings begrænsningerne er stadig valide, så dem kan vi blot bruge igen. Det vil sige, at den nye model bliver:

$$\begin{aligned}
&\max \sum_{i \in I} y_i \\
&s.t.: \sum_{j \in I} x_{0j} = 1 \\
&\sum_{i \in I} x_{i0} = 1 \\
&\sum_{i \in I_0} x_{ij} = y_j, \quad \forall j \in I \\
&\sum_{j \in I} x_{ij} = y_i, \quad \forall i \in I \\
&\sum_{i \in I_0} \sum_{j \in I_0} (t_{ij} + \tau_i) x_{ij} \leq 12 \cdot 60 = 720 \\
&u_i - u_j + n x_{ij} \leq n - 1, \quad \forall i, j \in I \\
&y_i \in \{0,1\}, \quad \forall i \in I_0 \\
&x_{ij} \in \{0,1\}, \quad \forall i, j \in I_0 \\
&0 \leq u_i \leq n, \quad \forall i \in I
\end{aligned}$$

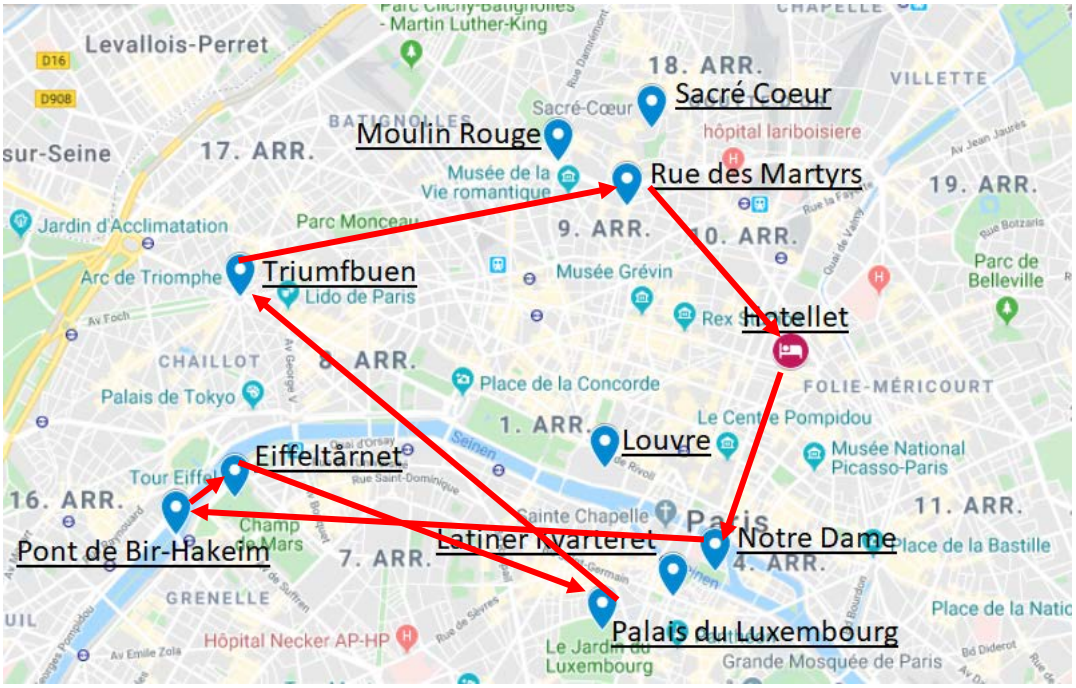
4. Implementer nu den nye model i OPL og løs den vha. CPLEX. Hvor mange seværdigheder kan Sofie nå at besøge på 12 timer? Illustrer den nye plan for besøget i Paris og kommenter på løsningens karakteristika såsom samlet tidsforbrug, rejsetid, rejseafstand osv..

Besvarelse:

Implementeres modellen fra opgave 3 i OPL fås følgende:

```
int n = ...;
range I = 1..n;
range I_0 = 0..n;

int t[I_0][I_0] = ...;
float d[I_0][I_0] = ...;
int tau[I_0] =...;

dvar boolean x[I_0][I_0];
dvar boolean y[I];
dvar float+ u[I] in I;

dexpr float TotalTravelTime = sum ( i, j in I_0 ) t[i][j] * x[i][j];
dexpr float TotalTime = sum ( i, j in I_0 ) ( tau[i] + t[i][j] ) * x[i][j];
dexpr float TotalDistance = sum ( i, j in I_0 ) d[i][j] * x[i][j];

maximize sum ( i in I ) y[i];
subject to
{
    sum ( j in I ) x[0][j] == 1;
    sum ( i in I ) x[i][0] == 1;
    forall ( i in I ) sum ( j in I_0 ) x[i][j] == y[i];
    forall ( j in I ) sum ( i in I_0 ) x[i][j] == y[j];

    sum ( i, j in I_0 ) (tau[i] + t[i][j] ) * x[i][j] <= 12*60;

    TotalTravelTime >= 0;
    TotalTime >= 0;
    TotalDistance >= 0;

    forall ( i,j in I )
     {
        u[i] - u[j] + n * x[i][j] <= n - 1;
     }
}
```

En optimal løsning består af 6 seværdigheder, nemlig Notre Dame, Pont de Bir-Hakeim, Eiffeltårnet, Palais du Luxembourg, Triumfbuen, Rues des Martyrs. Sightseeing turen kommer nu til at tage 700 minutter, eller 11 timer og 40 minutter. Rejsetiden mellem seværdighederne og hotellet beløber sig til i alt 295 minutter, eller 4 timer og 55 minutter. Den samlede distances som Sofie tilbagelægger er 23.1 kilometer. Plottes ruten på kortet fås:



Hvilket er en ret besynderlig tur.

Når Sofie ser på ruten, som fremkom i Opgave 4, ser den ikke ud til at være ”TSP-optimal”. Det vil sige, den ser ikke ud til at være den korteste vej fra hotellet, rundt blandt de udvalgte seværdigheder og så tilbage til hotellet.

5. Forklar med ord hvorfor det kan ske, at ruten ikke er optimal når vi har maksimeret antallet af seværdigheder som nås inden for de 12 timer.
   Besvarelse:
   Givet at objektfunktionen slet ikke har fokus på den samlede afstand eller tiden brugt på at rejse, men blot forsøger at klemme så mange besøg ind på tolv timer som muligt, er det ret naturligt, at selve ruten som vores solver finder, ikke er TSP optimal. Den har hurtigt fundet ud af, at den ikke kan putte mere end 6 besøg ind, og da den fandt en rute som tog mindre en 12 timer og som besøgte 6 seværdigheder, kunne den stoppe med den rute der nu engang var fundet.

Har du **ikke fået løst Opgave 4**, kan du fra nu antage, at det er følgende seks seværdigheder, som blev udvalgt i Opgave 4: Rue des Martyrs, Triumfbuen, Eiffeltårnet, Pont de Bir-Hakeim, Palais du Luxembourg og Notre Dame.

6. Find en optimal TSP tur gennem de udvalgte seværdigheder. Illustrer løsningen og sammenlign med løsningen fra Opgave 4. Beskriv ligeledes den nye rutes karakteristika.
   Besvarelse:
   For at kunne finde en optimal TSP tur genne de seværdigheder vi har udvalgt i opgave 4 kan vi ændre en lille bitte smule i modellen fra denne opgave. Det første vi skal gøre er at sørge for, at objektfunktionen optimerer den tid vi bruger på at rejse mellem seværdighederne. Det vil sige, at den bliver:

$$\min \sum_{i \in I_0} \sum_{j \in I_0} t_{ij} x_{ij}$$

   Derudover skal vi blot sørge for, at alle de rigtige seværdigheder bliver besøgt, hvilket kan gøres ved at sætte $y_i = 1$ for $i \in \{Rue\ des\ Martyrs,\ Triumfbuen, Eiffeltårnet, Pont\ de\ Bir -$

$Hakeim, Palais\ du\ Luxemburg, Notre\ Dame\}$ og $y_i = 0$ for de andre seværdigheder. Det leder til modellen:

$$\min \sum_{i \in I_0} \sum_{j \in I_0} t_{ij} x_{ij}$$

$$s.t.: \sum_{j \in I} x_{0j} = 1$$

$$\sum_{i \in I} x_{i0} = 1$$

$$\sum_{i \in I_0} x_{ij} = y_j, \quad \forall j \in I$$

$$\sum_{j \in I} x_{ij} = y_i, \quad \forall i \in I$$

$$\sum_{i \in I_0} \sum_{j \in I_0} (t_{ij} + \tau_i) x_{ij} \leq 12 \cdot 60 = 720$$

$$y_i = 1, \quad \forall i \in \{Rue\ des\ Martyrs, Triumfbuen, Eiffeltårnet, Pont\ de\ Bir - Hakeim, Palais\ du\ Luxemburg, Notre\ Dame\}$$

$$u_i - u_j + n x_{ij} \leq n - 1, \quad \forall i, j \in I$$

$$y_i \in \{0,1\}, \quad \forall i \in I_0$$

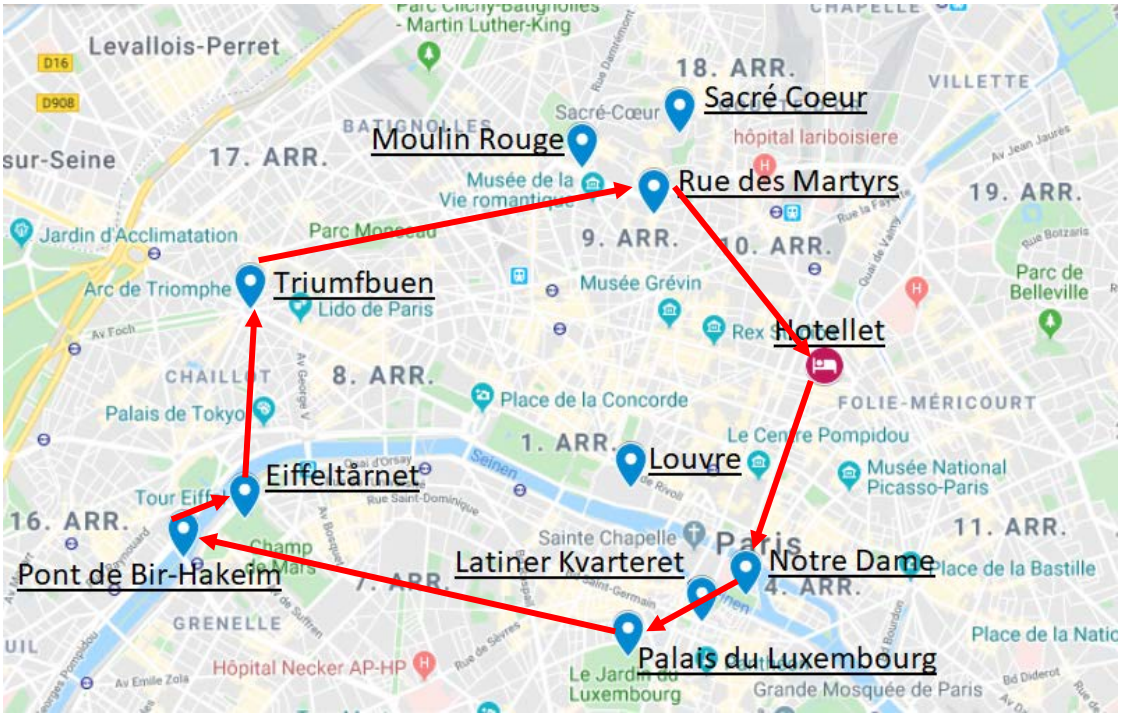
$$x_{ij} \in \{0,1\}, \quad \forall i, j \in I_0$$

$$0 \leq u_i \leq n, \quad \forall i \in I$$

Det leder til følgende implementering i OPL

```
int n = ...;
range I = 1..n;
range I_0 = 0..n;

int t[I_0][I_0] = ...;
float d[I_0][I_0] = ...;
int tau[I_0] =...;

dvar boolean x[I_0][I_0];
dvar boolean y[I];
dvar float+ u[I] in I;

dexpr float TotalTravelTime = sum ( i, j in I_0 ) t[i][j] * x[i][j];
dexpr float TotalTime = sum ( i, j in I_0 ) ( tau[i] + t[i][j] ) * x[i][j];
dexpr float TotalDistance = sum ( i, j in I_0 ) d[i][j] * x[i][j];

minimize sum ( i, j in I_0 ) t[i][j] * x[i][j];
subject to
{
    sum ( j in I ) x[0][j] == 1;
    sum ( i in I) x[i][0] == 1;
    forall ( i in I ) sum ( j in I_0 ) x[i][j] == y[i];
    forall ( j in I ) sum ( i in I_0 ) x[i][j] == y[j];

    sum ( i, j in I_0 ) (tau[i] + t[i][j] ) * x[i][j] <= 12*60;

    TotalTravelTime >= 0;
    TotalTime >= 0;
    TotalDistance >= 0;

    forall ( i,j in I )
    {
        u[i] - u[j] + n * x[i][j] <= n - 1;
    }
    y[1]==1;
    y[2]==1;
    y[3]==1;
    y[4]==0;
    y[5]==0;
    y[6]==1;
    y[7]==0;
    y[8]==1;
    y[9]==0;
    y[10]==1;
}
```

Løses denne model fås en ny rute gennem de seks seværdigheder som ser ud som følger:



Det leder til en tur som er 16.7 kilometer lang, tager 10 timer og 25 minutter at komme igennem, og hvor hun sammenlangt skal bruge 220 minutter eller 3 timer og 40 minutter på transport.

Sofie kommer nu i tanke om, at det måske i virkeligheden ikke er smart at minimere transporttiden på sin tur. I virkeligheden kunne man måske udnytte de 12 timer der er afsat til sightseeing i Paris på en bedre måde. Efter at have tænkt over situationen, kommer Sofie i tanke om, at det måske kunne være interessant at *maksimere* den afstand som hun tilbagelægger på sin tur mellem de udvalgte seværdigheder. På den måde får hun, antageligt, set mere af Paris, samtidig med, at hun besøger de udvalgte seværdigheder inden for de 12 timer, som er til rådighed.

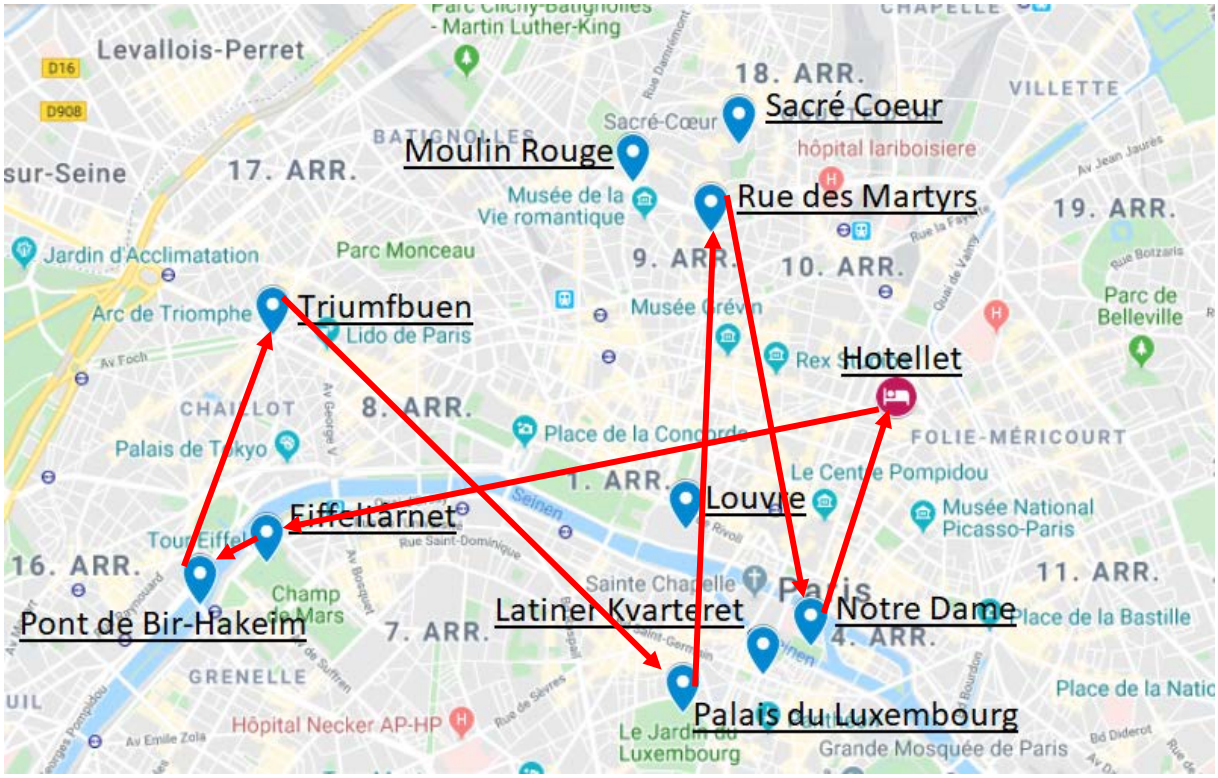
7. Ændr modellen fra Opgave 6, så Sofie ikke minimerer turens samlede tidsforbrug men i stedet maksimerer længden af turen under bibetingelse af, at dagens samlede program ikke må tage mere end 12 timer, og at alle de udvalgte seværdigheder bliver besøgt.
   Besvarelse:
   Det eneste der skal ændres i denne opgave i forhold til opgave 6 er, at man skal ændre objektfunktionen til

$$\max \sum_{i \in I_0} \sum_{j \in I_0} d_{ij} x_{ij}$$

   ellers er modellen fuldstændig identisk med den foregående.
   Det vil sige, at implementeringen kommer til at se således ud:

```
int n = ...;
range I = 1..n;
range I_0 = 0..n;

int t[I_0][I_0] = ...;
float d[I_0][I_0] = ...;
int tau[I_0] =...;

dvar boolean x[I_0][I_0];
dvar boolean y[I];
dvar float+ u[I] in I;

dexpr float TotalTravelTime = sum ( i, j in I_0 ) t[i][j] * x[i][j];
dexpr float TotalTime = sum ( i, j in I_0 ) ( tau[i] + t[i][j] ) * x[i][j];
dexpr float TotalDistance = sum ( i, j in I_0 ) d[i][j] * x[i][j];

maximize sum ( i, j in I_0 ) d[i][j] * x[i][j];
subject to
{
    sum ( j in I ) x[0][j] == 1;
    sum ( i in I ) x[i][0] == 1;
    forall ( i in I ) sum ( j in I_0 ) x[i][j] == y[i];
    forall ( j in I ) sum ( i in I_0 ) x[i][j] == y[j];

    TotalDistance >= 0;
    TotalTime >= 0;
    TotalTravelTime >= 0;

    sum ( i, j in I_0 ) (tau[i]+t[i][j] ) * x[i][j] <= 12*60;
    forall ( i,j in I )
     {
        u[i] - u[j] + n * x[i][j] <= n - 1;
     }
     y[1]==1;
    y[2]==1;
    y[3]==1;
    y[4]==0;
    y[5]==0;
    y[6]==1;
    y[7]==0;
    y[8]==1;
    y[9]==0;
    y[10]==1;
}
```

Løses modellen fås en løsning hvor Sofie skal gå 24.6 kilometer og hvor den samlede sightseeing tur tager præcis 12 timer. Endvidere skal hun bruge 315 minutter, eller 5 timer og 15 minutter på at transportere sig rundt i Paris. Den nye, lange, rute ser således ud:


Levallois-Perret
D16
D908
sur-Seine
17. ARR.
BATIGNOLLES
- Martin Luther-King
18. ARR.
Sacré Coeur
Sacré-Cœur
hôpital lariboisiere
VILLETTE
Moulin Rouge
Musée de la Vie romantique
Rue des Martyrs
19. ARR.
Jardin d'Acclimatation
Parc Monceau
9. ARR.
10. ARR.
Arc de Triomphe
Triumfbuen
Lido de Paris
Musée Grévin
Parc de Belleville
Hotellet
CHAILLOT
8. ARR.
Palais de Tokyo
Place de la Concorde
FOLIE-MÉRICOURT
1. ARR.
Le Centre Pompidou
Louvre
Eiffeltarnet
Musée National Picasso-Paris
Tour Eiffel
16. ARR.
11. ARR.
Sainte Chapelle
Latiner Kvarteret
Notre Dame
Pont de Bir-Hakeim
Champ de Mars
7. ARR.
4. ARR.
Place de la Bastille
GRENELLE
Palais du Luxembourg
Hôpital Necker AP-HP
Le Jardin du Luxembourg
Grande Mosquée de Paris