Treball de Fi de Grau

Grau en enginyeria en tecnologies industrials

Estudi de la producció de diversos tipus de peces en una línia de banys amb transport per grua

MEMÒRIA

Autor: Eva Moragues Escalona
Director: Manel Mateo Doll

Convocatòria: Setembre 2017

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

1. Algorisme

A continuació s'escriurà l'algorisme desenvolupat en aquest treball junt amb tots els subprogrames.

Algorisme

```
import math
from seqinicial import seqinicial
from EiF2 import EiF
from CalculV2 import CalculV2
from CalculLiU3 import calculLiU
from triarvertex2 import escollirvertex
from segonacomp3 import calculiteracions
from calculramificacions import calculramificacions
from CalculOrdre import CalculOrdre
from CalculVProv2 import CalculVProv2
from CalculU2 import CalculU
from Compfinal import compfinal

from sympy import*
y=Symbol('y')
import timeit

print("intro nombre de tancs m")
m=input(int)
print("intro nombre de peces n")
n=input(int)
print ("intro temps sense carrega e")
e=input(int)
print ("intro temps amb carrega f")
f=input(int)


A=[None for i in range ((m+1)*n)]
B=[None for i in range ((m+1)*n)]
z=0
while z<(m+1)*n:
    j=0
    for j in range(m+1):
        i=0
        for i in range(n):
            print("intro min obj",i, "en tanc",j)
            A[z]=input(int)
            print("intro max obj",i, "en tanc",j)
            B[z]=input(int)
            z=z+1
print("A=",A)
```

```
print("B=",B)

duplicat=0
dup=0
while dup<len(A):
    if A[dup]>duplicat:
        duplicat=A[dup]
        Dup=dup
    dup=dup+1

E,F=EiF(m,n,e,f,Dup)


tanc=Dup/n


H=[None for i in range (n*2)]
ll=[[None for j in range (7)] for k in range (3)]
ll[0][0]=H
ll=seqinicial(ll,n,Dup)
Lmax=(m+1)*n
g=0
if Dup==2 or Dup==3:
    while g<3:
        HProv=ll[g][0]
        V=CalculV2(m,n,E,F,e,f,ll,g,A,B)
        VProv=V
        UProv=[None for i in range (m+1)]
        UProv[1]=CalculU(VProv,HProv,n)
        ll=calculLiU(V,ll,g,n,UProv,m)
        if ll[g][1]==None:
            ll[g][1]=1
        i=2
        while i<=m:
            HProv=[None for j in range(n*(i+1))]
            HProv=CalculOrdre(i,n)
            VProv=CalculVProv2(i,n,E,F,e,f,A,B,HProv)
            UProv[i]=CalculU(VProv,HProv,n)
            ll[g][1]=1
            i=i+1

        g=g+1
    g=0
else:
    HProv=ll[0][0]
    V=CalculV2(m,n,E,F,e,f,ll,g,A,B)
    VProv=V
    UProv=[None for i in range (m+1)]
    UProv[1]=CalculU(VProv,HProv,n)
    ll=calculLiU(V,ll,g,n,UProv,m)
    ll[0][1]=1
    i=2
    while i<=m:
        HProv=[None for j in range(n*(i+1))]
```

```
        HProv=CalculOrdre(i,n)
        VProv=CalculVProv2(i,n,E,F,e,f,A,B,HProv)
        UProv[i]=CalculU(VProv,HProv,n)
        i=i+1
acabat=0
ultimaram=1

while acabat==0:
    if m!=1:
        aux=ll[0][0][:]
        aux2=ll[0][:]
        (g,acabat)=escollirvertex(ll,g,n,m,ultimaram)
        ultimaram=g
        if acabat==0:
            if len(ll[g][0])<m*n+n:
                ll=calculramificacions(V,ll,g,m,n,Dup,tanc)
                i=1
                while ll[i][0][0]!=None:
                    if ll[i][1]==None or ll[i][1]==2:
                        g=i
                        V=CalculV2(m,n,E,F,e,f,ll,g,A,B)
                        ll=calculLiU(V,ll,g,n,UProv,m)
                        if ll[g][1]==2:
                            (ll,r)=calculiteracions(V,ll,g,n,Dup)
                            if r!=0:
                                ll=compfinal(r,V,ll,g)

                    i=i+1
            else:
                (g,acabat)=escollirvertex(ll,g,n,m,ultimaram)

if m==1:
    ll[0][3]=0
    ll[0][4]=0
p=0
opt=0
while ll[p][1]!=None:
    print(ll[p])
    if len(ll[p][0])==m*n+n and ll[p][1]==1:
        if opt>ll[p][2] or opt==0:
            opt=ll[p][2]
            g=p
        else:
            pass

    p=p+1

print("la seq optima es:", ll[g])

def EiF (m,n,e,f,Dup):

   m=m+1
   E=[[None]*(m*n+n)for j in range (m*n+n)]
   F=[[None]*(m*n+n)for j in range (m*n+n)]
```

```
    if (m<n):
        p=0
        while p<=m:
            i=0
            auxiliar=1
            while i<n:
                aux=0
                while aux<n:
                    j=0
                    while j<n and (n*i+j)<n*auxiliar and (n*i+j+p*n)<m*n+n:
                        if(Dup%n!=0):
                            if((n*i+aux<Dup and n*i+p*n+j<Dup) or (n*i+aux>Dup and
n*i+p*n+j>Dup) or (n*i+aux==Dup and n*i+p*n+j==Dup)):
                                E[n*i+aux][n*i+p*n+j]=e*p
                                E[n*i+p*n+j][n*i+aux]=e*p
                                F[n*i+aux][n*i+p*n+j]=f*p
                                F[n*i+p*n+j][n*i+aux]=f*p
                            else:
                                E[n*i+aux][n*i+p*n+j]=e*(p+1)
                                E[n*i+p*n+j][n*i+aux]=e*(p+1)
                                F[n*i+aux][n*i+p*n+j]=f*(p+1)
                                F[n*i+p*n+j][n*i+aux]=f*(p+1)
                        else:
                            if((n*i+aux<Dup+1 and n*i+p*n+j<Dup+1) or (n*i+aux>Dup+1 and
n*i+p*n+j>Dup+1) or (n*i+aux==Dup+1 and n*i+p*n+j==Dup+1)):
                                E[n*i+aux][n*i+p*n+j]=e*p
                                E[n*i+p*n+j][n*i+aux]=e*p
                                F[n*i+aux][n*i+p*n+j]=f*p
                                F[n*i+p*n+j][n*i+aux]=f*p
                            else:
                                E[n*i+aux][n*i+p*n+j]=e*(p+1)
                                E[n*i+p*n+j][n*i+aux]=e*(p+1)
                                F[n*i+aux][n*i+p*n+j]=f*(p+1)
                                F[n*i+p*n+j][n*i+aux]=f*(p+1)
                        j=j+1
                    aux=aux+1
                i=i+1
                auxiliar=auxiliar+1
            p=p+1


    if (m==n):
        print('m=n')
        p=0
        while p<=n:
            i=0
            auxiliar=1
            while i<=n:
                aux=0
                while aux<n:
                    j=0
```

```
            while j<n and (n*i+p*n+j)<m*n+n:
               if(Dup%n!=0):
                  if((n*i+aux<Dup and n*i+p*n+j<Dup) or (n*i+aux>Dup and
n*i+p*n+j>Dup) or (n*i+aux==Dup and n*i+p*n+j==Dup)):
                     E[n*i+aux][n*i+p*n+j]=e*p
                     E[n*i+p*n+j][n*i+aux]=e*p
                     F[n*i+aux][n*i+p*n+j]=f*p
                     F[n*i+p*n+j][n*i+aux]=f*p
                  else:
                     E[n*i+aux][n*i+p*n+j]=e*(p+1)
                     E[n*i+p*n+j][n*i+aux]=e*(p+1)
                     F[n*i+aux][n*i+p*n+j]=f*(p+1)
                     F[n*i+p*n+j][n*i+aux]=f*(p+1)
               else:
                  if((n*i+aux<Dup+1 and n*i+p*n+j<Dup+1) or (n*i+aux>Dup+1 and
n*i+p*n+j>Dup+1) or (n*i+aux==Dup+1 and n*i+p*n+j==Dup+1)):
                     E[n*i+aux][n*i+p*n+j]=e*p
                     E[n*i+p*n+j][n*i+aux]=e*p
                     F[n*i+aux][n*i+p*n+j]=f*p
                     F[n*i+p*n+j][n*i+aux]=f*p
                  else:
                     E[n*i+aux][n*i+p*n+j]=e*(p+1)
                     E[n*i+p*n+j][n*i+aux]=e*(p+1)
                     F[n*i+aux][n*i+p*n+j]=f*(p+1)
                     F[n*i+p*n+j][n*i+aux]=f*(p+1)

               j=j+1
            aux=aux+1
         i=i+1
         auxiliar=auxiliar+1
      p=p+1

   if (m>n):
      print('m>n')
      p=0
      while p<=m:
         i=0
         auxiliar=1
         while i<=m:
            aux=0
            while aux<n:
               j=0
               while j<n and (n*i+p*n+j)<(m*n+n):
                  if(Dup%n!=0):
                     if((n*i+aux<Dup and n*i+p*n+j<Dup) or (n*i+aux>=Dup and
n*i+p*n+j>Dup) or (n*i+aux==Dup and n*i+p*n+j==Dup)):
                        E[n*i+aux][n*i+p*n+j]=e*p
                        E[n*i+p*n+j][n*i+aux]=e*p
                        F[n*i+aux][n*i+p*n+j]=f*p
                        F[n*i+p*n+j][n*i+aux]=f*p
                     else:
                        E[n*i+aux][n*i+p*n+j]=e*(p+1)
                        E[n*i+p*n+j][n*i+aux]=e*(p+1)
                        F[n*i+aux][n*i+p*n+j]=f*(p+1)
```

```
                    F[n*i+p*n+j][n*i+aux]=f*(p+1)
                else:
                    if((n*i+aux<Dup+1 and n*i+p*n+j<Dup+1) or (n*i+aux>=Dup+1 and
n*i+p*n+j>Dup+1) or (n*i+aux==Dup+1 and n*i+p*n+j==Dup+1)):
                        E[n*i+aux][n*i+p*n+j]=e*p
                        E[n*i+p*n+j][n*i+aux]=e*p
                        F[n*i+aux][n*i+p*n+j]=f*p
                        F[n*i+p*n+j][n*i+aux]=f*p
                    else:
                        E[n*i+aux][n*i+p*n+j]=e*(p+1)
                        E[n*i+p*n+j][n*i+aux]=e*(p+1)
                        F[n*i+aux][n*i+p*n+j]=f*(p+1)
                        F[n*i+p*n+j][n*i+aux]=f*(p+1)

                j=j+1
            aux=aux+1
        i=i+1
        auxiliar=auxiliar+1
    p=p+1

EIF=[E,F]
return EIF
```

def seqinicial(ll,n):

```
t=0
d=0
while t<n:
    q=0
    while q<2:
        ll[0][0][d]=t+n*q
        q=q+1
        d=d+1
    t=t+1

return ll[0]
```

def calculV(m,n,E,F,e,f,ll,g,A,B):

```
from sympy import symbols
from sympy.abc import y


V=[[[None for k in range (2)] for j in range (len(ll[g][0]))] for i in range (len(ll[g][0]))]
i=len(ll[g][0])-1
x=ll[g][0][i]
V[x][0][0]=f+e+E[x][0]-y

if m==1:
```

```
    j=0
    t=0
    x=0
    i=0
    while x<n:
        if V[i*n+t][n+t+i*n][0]!=None or V[n+t+i*n][i*n+t][0]!=None:
            try:
                V[i*n+t][n+t+i*n-1][1]=F[i*n+t][n+t+i*n-1]+A[i*n+t+n]
                V[n+t+i*n-1][i*n+t][1]=-(F[i*n+t][n+t+i*n-1]+B[i*n+t+n])
            except:
                pass

        else:
            try:
                V[i*n+t][n+t+i*n][0]=F[i*n+t][n+t+i*n]+A[i*n+t+n]
                V[n+t+i*n][i*n+t][0]=-(F[i*n+t][n+t+i*n]+B[i*n+t+n])
            except:
                pass


        j=j+1
        t=t+1
        x=x+1


else:

    i=0
    while i<len(ll[g][0]):
        j=i+1
        while j<len(ll[g][0]):
            if ll[g][0][i]==ll[g][0][j]+n:
                x=ll[g][0][i]
                z=ll[g][0][j]
                if V[x][z][0]!=None:
                    V[z][x][1]=F[x][z]+A[x]-y
                    V[x][z][1]=-F[z][x]-B[x]+y

                else:
                    V[z][x][0]=F[x][z]+A[x] - y
                    V[x][z][0]=-F[z][x]-B[x] + y

            j=j+1
        i=i+1


    i=0
    if n==1 or m==1:
        b=1
    else:
        b=n
    while i<((len(ll[g][0])-b)/b):
        j=0
        t=0
```

```
            x=0
            while x<n:
               if V[i*n+t][n+t+i*n][0]!=None or V[n+t+i*n][i*n+t][0]!=None:
                  try:
                     V[i*n+t][n+t+i*n][1]=F[i*n+t][n+t+i*n]+A[i*n+t+n]
                     V[n+t+i*n][i*n+t][1]=-(F[i*n+t][n+t+i*n]+B[i*n+t+n])
                  except:
                     pass

               else:
                  try:
                     V[i*n+t][n+t+i*n][0]=F[i*n+t][n+t+i*n]+A[i*n+t+n]
                     V[n+t+i*n][i*n+t][0]=-(F[i*n+t][n+t+i*n]+B[i*n+t+n])
                  except:
                     pass


               j=j+1
               t=t+1
               x=x+1
            i=i+1


      i=0
      while i<(len(ll[g][0])-1):
         x=ll[g][0][i]
         z=ll[g][0][i+1]
         if V[x][z][0]==None:
            try:
               V[x][z][0] = F[x][x+n] + E[x+n][z]
            except:
               pass

         else:
            try:
               V[x][z][1] = F[x][x+n] + E[x+n][z]
            except:
               pass

         i=i+1

      return V
```

def calculLiU (V,ll,g,n,UProv,m):

```
      from sympy import*
      y=Symbol('y')
      import math
      H=ll[g][0]
      L=[None for i in range (len(H))]
      L[0]=0
      Mult=0
      i=0
```

```
    for i in range(len(H)-1):
        x=H[i]
        z=H[i+1]

        try:
            solve(V[x][z][0],y)
            L[0]=L[0]+V[x][z][0]
        except:
            L[0]=L[0]+V[x][z][0]


    i=len(H)-1
    x=H[i]

    try:
        solve(V[x][0][0],y)
        L[0]=L[0]+V[x][0][0]
    except:
        L[0]=L[0]+V[x][0][1]



        i=0
        while i<len(H)-1:
            j=i+1
            x=H[i]
            z=H[j]
            primer=0
            aux=z
            if x>z and x!=z+n:
                auxx=x
                auxz=z
                L[aux]=0
                L[aux]=L[aux]+V[x][z][0]
                trobat=0
                trobat2=0
                opcio=0
                mult=1
                Mult=0

                while trobat==0:
                    if j==len(H)-1 and trobat==0:
                        while x-n*mult>=0 and trobat ==0 and Mult==0 or x+n*mult<=len(H) and
trobat==0 and Mult==0:
                            if H[j]==x-n*mult and H[j]<x:
                                trobat=1
                                opcio=1

                            elif H[j]==x+n*mult and H[j]<x:
                                trobat=1
                            else:
                                mult=mult+1
                            if trobat==1:
                                Mult=mult
```

```
                mult=1
                if trobat==0:
                    L[aux]=L[aux]+V[H[j]][0][0]
                    j=0
                    mult=1
                    if H[1]==x:
                        L[aux]=L[aux]+V[0][x][0]
                        trobat=1
                        trobat2=1
                    while x-n*mult>=0 and trobat ==0 and Mult==0 or x+n*mult<=len(H) and
trobat==0 and Mult==0:
                        if H[j]==x-n*mult and H[j]<x:
                            trobat=1
                            opcio=1

                        elif H[j]==x+n*mult and H[j]<x:
                            trobat=1
                        else:
                            mult=mult+1
                        if trobat==1:
                            Mult=mult


            else:
                while x-n*mult>=0 and trobat ==0 and Mult==0 or x+n*mult<=len(H) and
trobat==0 and Mult==0:
                    if H[j]==x-n*mult and H[j]<x:
                        trobat=1
                        opcio=1

                    elif H[j]==x+n*mult and H[j]<x:
                        trobat=1

                    else:
                        mult=mult+1
                    if trobat==1:
                        Mult=mult

                if primer==0 and z!=H[len(H)-1] and trobat==0:
                    L[aux]=L[aux]+V[z][H[j+1]][0]
                    primer=1

                if j+1==len(H)-1 and trobat==0:
                    L[aux]=L[aux]+V[H[j+1]][0][0]
                    j=0
                    mult=1
                    if H[1]==x:
                        L[aux]=L[aux]+V[0][x][0]
                        trobat=1
                        trobat2=1
                    while x-n*mult>=0 and trobat==0 and Mult==0 or x+n*mult<=len(H) and
trobat==0 and Mult==0:
                        if H[j]==x-n*mult and H[j]<x:
                            trobat=1
```

```
                    opcio=1
                elif H[j]==x+n*mult and H[j]<x:
                    trobat=1
                else:
                    mult=mult+1
                if trobat==1:
                    Mult=mult

            mult=1
            while x-n*mult>=0 and trobat==0 and Mult==0 and H[j]!=0 or
x+n*mult<=len(H) and trobat==0 and Mult==0 and H[j]!=0:
                if H[j+1]==x-n*mult and H[j+1]<x:
                    trobat=1
                    opcio=1

                elif H[j+1]==x+n*mult and H[j+1]<x:
                    trobat=1

                else:
                    mult=mult+1
                if trobat==1:
                    Mult=mult
            if trobat==0 and trobat2==0:
                if H[j]==0:
                    L[aux]=L[aux]+V[H[j]][H[j+1]][0]
                    mult=1
                else:
                    L[aux]=L[aux]+V[H[j+1]][H[j+2]][0]
                    mult=1

                if H[j+2]==x:
                    trobat=1
                    trobat2=1
                j=j+1
        if trobat==1 and trobat2==0:
            while Mult>0:
                if opcio==1:
                    L[aux]= L[aux]+ V[x-n*Mult][x-n*(Mult-1)][0]

                else:
                    L[aux]=L[aux]+ V[x+n*Mult][x+n*(Mult-1)][0]

                Mult=Mult-1

        else:
            j=j+1


    i=i+1
i=len(H)-1
x=H[i]
z=0
j=0
Laux=0
```

```
    Laux=Laux+V[x][z][0]
    trobat=0
    mult=1
    opcio=0
    while x-n*mult>=0 and trobat==0 or x+n*mult<=len(H) and trobat==0:
        if H[j+1]==x-n*mult and H[j+1]<x:
            trobat=1
            opcio=1

        elif H[j+1]==x+n*mult and H[j+1]<x:
            trobat=1
        else:
            mult=mult+1
        if trobat==1:
            Mult=mult

        if trobat==0 and x-n*mult<0 and x+n*mult>len(H):
            Laux=Laux + V[H[j]][H[j+1]][0]
            mult=1
            j=j+1


    if trobat==1:
        if opcio==1:
            Laux=Laux+V[H[j]][x-n*Mult][0]
            while Mult>0:
                Laux= Laux+ V[x-n*Mult][x-n*(Mult-1)][0]
                Mult=Mult-1

        else:
            Laux=Laux+V[H[j]][x+n*Mult][0]
            while Mult>0:
                Laux=Laux+ V[x+n*Mult][x+n*(Mult-1)][0]
                Mult=Mult-1


    if L[x]==None:
        L[x]=Laux
    else:
        lx=solve(L[x],y)
        laux=solve(Laux,y)
        if lx>=laux:
            pass
        else:
            L[x]=Laux



U=[None for i in range (len(H))]
if n==1:

    i=0
    while i<len(H)-1:
        j=i+1
```

```
            x=H[i]
            z=H[j]
            aux=z
            if x+n<=z:
               U[aux]=0
               if V[x][z][0]!=None:

                  U[aux]=U[aux]+V[x][z][0]
                  l=1
                  while (z-l*n)>=x:
                     if (z-l*n)==0 and V[z-(l-1)*n][z-l*n][1]!=None:

                        U[aux]=U[aux]+V[z-(l-1)*n][z-l*n][1]

                     else:
                        U[aux]=U[aux]+V[z-(l-1)*n][z-l*n][0]


                     l=l+1

            i=i+1
      else:
         rep=0
         i=0
         while i<len(H)-1:
            j=i+1
            x=H[i]
            z=H[j]
            primer=0
            aux=z
            if x<z and z!=x+n:
               auxx=x
               auxz=z
               U[aux]=0
               U[aux]=U[aux]+V[x][z][0]
               trobat=0
               trobat2=0
               opcio=0
               mult=1
               Mult=0
               while trobat==0:

                  if j==len(H)-1 and trobat==0:
                     while x-n*mult>=0 and trobat ==0 and Mult==0 or x+n*mult<=len(H) and
trobat==0 and Mult==0:
                        if H[j]==x-n*mult and H[j]>x:
                           trobat=1
                           opcio=1

                        elif H[j]==x+n*mult and H[j]>x:
                           trobat=1
                        else:
                           mult=mult+1
                        if trobat==1:
```

```
                    Mult=mult


                mult=1
                if trobat==0:
                    U[aux]=U[aux]+V[H[j]][0][0]
                    j=0
                    mult=0

                    if H[1]==x:

                        U[aux]=U[aux]+V[0][x][0]
                        trobat=1
                        trobat2=1
                    else:
                        U[aux]=U[aux]+V[0][H[j+1]][0]
                    while x-n*mult>=0 and trobat ==0 and Mult==0 or x+n*mult<=len(H) and
trobat==0 and Mult==0:
                        if H[j]==x-n*mult and H[j]>x:
                            trobat=1
                            opcio=1

                        elif H[j]==x+n*mult and H[j]>x:
                            trobat=1
                        else:
                            mult=mult+1
                        if trobat==1:
                            Mult=mult

            else:
                mult=1
                while x-n*mult>=0 and trobat==0 and Mult==0 or x+n*mult<=len(H) and
trobat==0 and Mult==0:
                    if H[j]==x-n*mult and H[j]>x:
                        trobat=1
                        opcio=1

                    elif H[j]==x+n*mult and H[j]>x:
                        trobat=1

                    else:
                        mult=mult+1
                    if trobat==1:
                        Mult=mult
                if primer==0 and z!=H[len(H)-1] and trobat==0:
                    U[aux]=U[aux]+V[z][H[j+1]][0]
                    primer=1


                if j+1==len(H)-1 and trobat==0:
                    U[aux]=U[aux]+V[H[j+1]][0][0]
                    j=-1
                    mult=1
                    if H[1]==x:
```

```
                    L[aux]=L[aux]+V[0][x][0]
                    trobat=1
                    trobat2=1


                while x-n*mult>=0 and trobat==0 and Mult==0 or x+n*mult<=len(H) and
trobat==0 and Mult==0:
                    if H[j+1]==x-n*mult and H[j+1]>x:
                        trobat=1
                        opcio=1

                    elif H[j+1]==x+n*mult and H[j+1]>x:
                        trobat=1

                    else:
                        mult=mult+1
                    if trobat==1:
                        Mult=mult


            mult=1
            while x-n*mult>=0 and trobat==0 and Mult==0 or x+n*mult<=len(H) and
trobat==0 and Mult==0:
                if H[j+1]==x-n*mult and H[j+1]>x:
                    trobat=1
                    opcio=1

                elif H[j+1]==x+n*mult and H[j+1]>x:
                    trobat=1

                else:
                    mult=mult+1
                if trobat==1:
                    Mult=mult


            if trobat==0 and trobat2==0 and j+1!=len(H)-1:


                if H[j]==0 and primer==0:
                    U[aux]=U[aux]+V[H[j]][H[j+1]][0]

                else:
                        U[aux]=U[aux]+V[H[j+1]][H[j+2]][0]
                        mult=1



                if H[j]==x:
                    trobat=1
                    trobat2=1
                j=j+1

        if trobat==1 and trobat2==0:
```

```
            while Mult>0:
                if opcio==1:

                    U[aux]= U[aux]+ V[x-n*Mult][x-n*(Mult-1)][0]

                else:
                    U[aux]=U[aux]+ V[x+n*Mult][x+n*(Mult-1)][0]

                Mult=Mult-1


        else:
            j=j+1


        novaUN=False
        tu=1
        UN=V[auxx][auxz][0]
        while auxz-tu*n>auxx and novaUN==False:
            TU=1
            UN=UN+V[auxz-(tu-1)*n][auxz-tu*n][0]
            while auxx+TU*n<auxz and novaUN==False:
                if V[auxz-tu*n][auxx+TU*n][0]!=None:
                    UN=UN+V[auxz-tu*n][auxx+TU*n][0]
                    novaUN=True
                    while auxx+(TU-1)*n>=auxx:
                        UN=UN+V[auxx+TU*n][auxx+(TU-1)*n][0]
                        TU=TU-1

                else:
                    TU=TU+1
            tu=tu+1
        if novaUN==True:
            UNtemp=solve(UN,y)
            if U[aux]!=None:
                Utemp=solve(U[aux],y)
            else:
                Utemp=0
            if UNtemp>Utemp or U[aux]==None or Utemp==[]:
                U[aux]=UN
            else:
                pass

    i=i+1


i=0
while i<len(U):
    if U[i]!=None:
        try:
            if y in U[i]:
                s=solve(U[i],y)
                U[i]=s
        except:
```

```
            pass

        else:
            pass
        i=i+1


    UB=None
    i=0
    while i<len(H):
        if U[i]!=None:
            try:
                U[i]=solve(U[i],y)
                if U[i][0]>=0 or U[i][0]<0:
                    pass
                else:
                    U[i]=None
            except:
                U[i]=None
            if U[i]!=None:
                if UB==None:
                    UB=U[i]
                elif UB<U[i]:
                    pass
                else:
                    UB=U[i]
        i=i+1

    LB=None
    i=0
    while i<len(H):
        if L[i]!=None:
            try:
                L[i]=solve(L[i],y)
            except:
                L[i]=None
            if L[i]!=None:
                if LB==None:
                    LB=L[i]
                elif LB>L[i]:
                    pass
                else:
                    LB=L[i]

        i=i+1

    ll[g][2]=LB[0]


    if UB!=None:
        ll[g][6]=UB[0]

    if UB==None:
        UB=solve(UProv[((len(ll[g][0]))/n)-1],y)
```

```
        ll[g][6]=UB[0]

    if LB<=UB:
        pass
    elif LB>UB:
        ll[g][1]=0

    return ll
```

def calculramificacions(V,ll,g,m,n, Dup):

```
def calculramificacions(V,ll,g,m,n,Dup,tanc):
    aux=[None for j in range (7)]
    aux2=[None for j in range (7)]
    aux[0]=ll[g][0][:]
    aux2[0]=ll[g][:]

    if Dup==2 or Dup==3:
        aux[1]=ll[1][0][:]
        aux2[1]=ll[1][:]
        aux[2]=ll[2][0][:]
        aux2[2]=ll[2][:]
    import math

    if g==0:

        if m!=1 and n!=1:
            if m*n<7:
                p=math.factorial(n*m)
            else:
                p=math.factorial(9)

        if m==1:
             p=1

        if n==1:
            p=math.factorial(m+1)


        H=[None for i in range (n*2)]
        ll=[[None for j in range (7)]for k in range (p)]
        if Dup==2 or Dup==3:
            i=3
            x=3
            while x>0:
                #print(i-x)
                ll[i-x][0]=aux[i-x]
                ll[i-x]=aux2[i-x]
                ll[i-x][3]=0
                ll[i-x][4]=0
                x=x-1
```

```
            while i<p:
                ll[i][0]=H
                i=i+1
            ll[0][5]=1

        else:
            i=1

            while i<p:
                ll[i][0]=H
                i=i+1
            ll[0][0]=aux[0]
            ll[0]=aux2[0]
            ll[0][5]=1
            ll[0][3]=0
            ll[0][4]=0
else:
    G=g
    while ll[G][0][0]!=None:
        G=G+1
    ll[G][3]=ll[g][3]+1
    ll[G][4]=ll[g][3]

w=1
t=len(ll[g][0])
ll[g][5]=1
q=t
j=1
if n==1:
    while j<=t:
        if ll[w][0][0]==None:
            ll[w][0]=aux[:]
            ll[w][0].insert(j,q)
            ll[w][1]=2
            ll[w][4]=ll[0][3]
            ll[w][3]=ll[0][3]+1
            j=j+1

        w=w+1

else:
    while j<=t and w<len(ll)-1:
        if ll[w][0][0]==None:
            if ll[w][1]==None:
                if ll[w+1][0][0]!=None:
                    w=w+1
                else:
                    ll[w][0]=aux[0][:] #cambiar por aux[] apropiado!!!!
                    ll[w][0].insert(j,q)
                    ll[w][4]=ll[g][3]
                    ll[w][3]=ll[g][3]+1
                    j=j+1

        w=w+1
```

```
w=w-1
t=len(ll[w][0])
q=t
auxiliar=0
w=1
j=n

while ll[w][0][0]!=None:
    if n==1:
        break


    else:
        if len(ll[w][0])==t and ll[w][1]==None:
            pos=ll[w][3]
            pos2=ll[w][4]
            ll[w][1]=0
            aux=ll[w][0][:]
            trobat=0
            while trobat==0:
                if ll[w+1][0][0]!=None and ll[w+1][1]!=0:
                    w=w+1
                else:
                    trobat=1
                    l=1
                    while l<=t and w<len(ll)-1:
                        ll[w+1][0]=aux[:]
                        ll[w+1][0].insert(l,t)
                        ll[w+1][3]=pos
                        ll[w+1][4]=pos2
                        L=1
                        while L<t:
                            if ((ll[w+1][0][L]/n==ll[w+1][0][L-1]/n and
tanc!=ll[w+1][0][L]/n and tanc-1!=ll[w+1][0][L]/n) or (ll[w+1][0][L]/n==ll[w+1][0][L+1]/n
and tanc!=ll[w+1][0][L]/n) and tanc-1!=ll[w+1][0][L]/n):
                                ll[w+1][1]=0
                            if ((ll[w+1][0][L]/n)+1)==(ll[w+1][0][L+1]/n) and
ll[w+1][0][L+1]!=ll[w+1][0][L]+n and tanc!=ll[w+1][0][L+1]/n:
                                ll[w+1][1]=0

                            L=L+1


                        T=2
                        len1=len(ll[w+1][0])
                        while T<=n and ll[w+1][1]!=0:
                            i=1
                            pos11=0
                            pos22=0
                            while i<len1:
                                if ll[w+1][0][i]==len1-T:
                                    pos11=i
```

```
                    if ll[w+1][0][i]==len1-T-n:
                        pos22=i


                    i=i+1

                if pos11<pos22 and ll[w+1][0][pos11]/n!=tanc:
                    if n==2:
                        if pos22!=len1-1 and pos22!=len1-2:
                            ll[w+1][1]=0


                    else:
                        ll[w+1][1]=0

                else:
                    pass
                T=T+1

            if ll[w+1][0][1]==len(ll[w+1][0])-1 and
ll[w+1][0][2]==ll[w+1][0][1]-n:
                ll[w+1][1]=0



        if ll[w+1][1]!=0: #aqui empiezan los cambios.14/02/16.
            i=1
            Factible=0
            while i<len(ll[w+1][0]) and Factible==0:
                k=i+1
                Found=0
                while k<len(ll[w+1][0]) and Factible==0:
                    if (ll[w+1][0][i]/n==ll[w+1][0][k]/n or
(ll[w+1][0][i]/n)+1==ll[w+1][0][k]/n) and ll[w+1][0][k]!=ll[w+1][0][i]+n and
ll[w+1][0][i]/n!=((len(ll[w+1][0])/n)-1) and tanc!=(ll[w+1][0][k]+n)/n and
tanc!=(ll[w+1][0][k])/n:

                        j=i+1
                        while j<len(ll[w+1][0]) and Found==0:
                            if ll[w+1][0][j]==ll[w+1][0][i]+n:
                                print("ll[w+1][0][j]",ll[w+1][0][j])
                                Found=1
                                if j>k:
                                    print("infactible por nueva parte", ll[w+1])

                                    Factible=1
                                    ll[w+1][1]=0
                            if Found==0 and j==len(ll[w+1][0])-1:
                                Factible=1
                                print("infactible por nueva parte2",ll[w+1])

                                ll[w+1][1]=0
```

```
                                        j=j+1
                                    k=k+1
                                i=i+1
                        l=l+1
                        if ll[w+1][1]==0:
                            ll[w+1][1]=None
                            pass
                        elif  w<len(ll)-2:
                            ll[w+1][1]=2
                            w=w+1
                        else:
                            break



                    i=len(ll[w+1][0])-1
                    while i>=0:
                        ll[w+1][0][i]=None
                        ll[w+1][1]=None
                        ll[w+1][3]=None
                        ll[w+1][4]=None
                        i=i-1
                    w=1

                else:
                    w=w+1
    return ll
```

def calculiteracions(V,ll,g,n,Dup):

```
from comprovacioincrement3 import comprovacioincrement
    from sympy import*
    y=Symbol('y')

    trobat1=2
    r=0
    Vsubst=[[[None for k in range (2)] for j in range (len(ll[g][0]))] for i in range (len(ll[g][0]))]
    i=0
    while i<len(ll[g][0]):
        j=0
        while j<len(ll[g][0]):
            Vsubst[i][j]=V[i][j][:]
            Vsubst[i][j]=V[i][j][:]
            j=j+1
        i=i+1

    H=ll[g][0]
    R=[[None for k in range(len(H))] for j in range(8)]
    L=ll[g][2]
    U=ll[g][6]
    TC=min(L,U)
```

```
t=[None for j in range (len(H))]
T=[None for j in range (len(H))]
t[0]=0

R[0][0]=0
d=0
Rec=[None for j in range (len(H))]
trobat=0
if ll[g][1]==2 or ll[g][1]==None:
   z=0
   while z<len(H):
      x=0
      while x<len(H):
         p=2
         if Vsubst[x][z][0]!=None:
            p=0
         else:
            if Vsubst[x][z][1]!=None:
                  p=1
            if n==1:
               if z==0:
                  p=0
               else:
                  p=1
         if (p==1 or p==0) and Vsubst[x][z][p]!=None:
            w1=solve(Vsubst[x][z][p],y)
            try:
               w1=w1[0]
               if w1<0:
                  w1=Vsubst[x][z][p]-w1
               else:
                  w1=Vsubst[x][z][p]+w1
               if w1-y!=0:
                  Vsubst[x][z][p]=Vsubst[x][z][p]+y
                  Vsubst[x][z][p]=Vsubst[x][z][p]-TC

               else:
                  Vsubst[x][z][p]=Vsubst[x][z][p]-y
                  Vsubst[x][z][p]=Vsubst[x][z][p]+TC

            except:
               pass
         x=x+1
      z=z+1
while trobat==0:
   z=0
   while z<len(H):
      x=0
      while x<len(H):

         if t[x]!=None and t[z]==None:
            if Vsubst[x][z][0]!=None:
               T[z]=t[x]+Vsubst[x][z][0]
               if t[z]==None or T[z]>t[z]:
```

```
                if T[z]>0:
                    t[z]=T[z]
                    R[0][z]=t[z]
                    if Rec[z]==None:
                        d=d+1

            if R[0][z]!=None and Rec[z]!=1:
                Rec[z]=1
            if d==len(H)-1:
                trobat=1
        x=x+1

    z=z+1


i=1
while i<8:
    d=0
    trobat=0
    Rec=[None for j in range (len(H))]
    T1=[None for j in range (len(H))]
    T=[None for j in range (len(H))]
    t=[None for j in range(len(H))]
    x=0
    z=0
    T[z]=0
    while x<len(H):
        if Vsubst[x][0][0]!=None:
            if Vsubst[x][0][1]!=None and Vsubst[x][0][1]>Vsubst[x][0][0]:
                T1[z]=R[i-1][x]+Vsubst[x][0][1]
                if T1[z]>T[z]:
                    T[z]=T1[z]

            else:
                T1[z]=R[i-1][x]+Vsubst[x][0][0]
                if T1[z]>T[z]:
                    T[z]=T1[z]

        x=x+1

    t=[None for j in range (len(H))]
    t[0]=T[0]
    R[i][0]=T[0]
    while trobat==0 and d<len(H):
        z=1
        while z<len(H):
            x=0
            while x<len(H):
                if Vsubst[x][z][0]!=None and t[x]!=None and T[z]==None:
                    if t[x]>=R[i-1][x]:
                        T[z]=t[x]+Vsubst[x][z][0]
                    else:
                        T[z]=R[i-1][x]+Vsubst[x][z][0]
                    w=0
```

```
                while w<len(H):
                    if Vsubst[w][z][0]!=None:
                        if R[i-1][w]+Vsubst[w][z][0]>T[z]:
                            T[z]=R[i-1][w]+Vsubst[w][z][0]
                    else:
                        pass
                    w=w+1

                if t[z]==None or T[z]>t[z]:
                    if T[z]>0:
                        t[z]=T[z]
                        R[i][z]=t[z]
                        if Rec[z]==None:
                            d=d+1
                if R[i][z]!=None:
                    Rec[z]=1


            if d==len(H)-1:
                    trobat=1
            x=x+1

        z=z+1

    i2=0
    while i2<len(R[i-1]) and trobat1!=0:
        if R[i-1][i2]==R[i][i2]:
            i2=i2+1
            if i2==len(H):
                trobat1=0
                break

        else:
            trobat1=1
            break
    i=i+1


if trobat1==0:
    ll[g][1]=1
    r=R[i-1]

else:
    op=[None for j in range(7)]
    trobat=0
    j=0
    while j<7:
        op[j]=R[j+1][0]-R[j][0]
        i=0
        while trobat==0 and i<len(R[j]):
            if R[j+1][i]-R[j][i]<op[j] or op[j]==0:
                if R[j+1][i]-R[j][i]!=0:
                    op[j]=R[j+1][i]-R[j][i]
```

```
            else:
                pass
            i=i+1
        j=j+1


    OP=op[0]
    i=0
    while i<6:
        if op[i+1]<op[i]:
            OP=op[i+1]
        i=i+1



    if TC+OP<=ll[g][6] and OP!=0:
        ll[g][2]=TC+OP
        (ll,r)=comprovacioincrement(V,ll,g,n)
    else:
        ll[g][1]=0


  return (ll,r)
```

def comprovacioincrement(V,ll,g,n):

```
  from sympy import*
  y=Symbol('y')

  Vsubst=[[[None for k in range (2)] for j in range (len(ll[g][0]))] for i in range (len(ll[g][0]))]
  H=ll[g][0]
  U=ll[g][6]
  FI=0
  while FI==0:
      i=0
      while i<len(ll[g][0]):
          j=0
          while j<len(ll[g][0]):
              Vsubst[i][j]=V[i][j][:]
              Vsubst[i][j]=V[i][j][:]
              j=j+1
          i=i+1
      L=ll[g][2]
      TC=min(L,U)
      t=[None for j in range (len(H))]
      T=[None for j in range (len(H))]
      t[0]=0
      R=[[None for k in range(len(H))] for j in range(10)]

      R[0][0]=0
      d=0
      Rec=[None for j in range (len(H))]
```

```
trobat=0
if ll[g][1]==2 or ll[g][1]==None:
   z=0
   while z<len(H):
      x=0
      while x<len(H):
         p=2
         if Vsubst[x][z][0]!=None:
            p=0
         else:
            if Vsubst[x][z][1]!=None:
                  p=1
            if n==1:
               if z==0:
                  p=0
               else:
                  p=1
         if (p==0 or p==1) and Vsubst[x][z][p]!=None:
            w1=solve(Vsubst[x][z][p],y)
            try:
               w1=w1[0]
               if w1<0:
                  w1=Vsubst[x][z][p]-w1
               else:
                  w1=Vsubst[x][z][p]+w1
               if w1-y!=0:
                  Vsubst[x][z][p]=Vsubst[x][z][p]+y
                  Vsubst[x][z][p]=Vsubst[x][z][p]-TC

               else:
                  Vsubst[x][z][p]=Vsubst[x][z][p]-y
                  Vsubst[x][z][p]=Vsubst[x][z][p]+TC

            except:
               pass
         x=x+1
      z=z+1
while trobat==0:
   z=0
   while z<len(H):
      x=0
      while x<len(H):

         if t[x]!=None and t[z]==None:
            if Vsubst[x][z][0]!=None:
               T[z]=t[x]+Vsubst[x][z][0]
               if t[z]==None or T[z]>t[z]:
                  if T[z]>0:
                     t[z]=T[z]
                     R[0][z]=t[z]
                     #print("R1",R1)
                     if Rec[z]==None:
                        d=d+1
```

```
                if R[0][z]!=None and Rec[z]!=1:
                    Rec[z]=1
                if d==len(H)-1:
                    trobat=1
            x=x+1

        z=z+1

d=0
trobat=0
Rec=[None for j in range (len(H))]
T1=[None for j in range (len(H))]
T=[None for j in range (len(H))]
t=[None for j in range(len(H))]
x=0
z=0
T[z]=0
while x<len(H):
    if Vsubst[x][0][0]!=None:
        if Vsubst[x][0][1]!=None and Vsubst[x][0][1]>Vsubst[x][0][0]:
            T1[z]=R[0][x]+Vsubst[x][0][1]
            if T1[z]>T[z]:
                T[z]=T1[z]
        else:
            T1[z]=R[0][x]+Vsubst[x][0][0]
            if T1[z]>T[z]:
                T[z]=T1[z]
    x=x+1
t=[None for i in range (len(H))]
t[0]=T[0]
R[1][0]=T[0]
while trobat==0 and d<len(H):
    z=1
    while z<len(H):
        x=0
        while x<len(H):
            if Vsubst[x][z][0]!=None and t[x]!=None and T[z]==None:
                if t[x]>=R[0][x]:
                    T[z]=t[x]+Vsubst[x][z][0]
                else:
                    T[z]=R[0][x]+Vsubst[x][z][0]
                w=0
                while w<len(H):
                    if Vsubst[w][z][0]!=None:
                        if R[0][w]+Vsubst[w][z][0]>T[z]:
                            T[z]=R[0][w]+Vsubst[w][z][0]
                    else:
                        pass
                    w=w+1

                if t[z]==None or T[z]>t[z]:
                    if T[z]>0:
                        t[z]=T[z]
                        R[1][z]=t[z]
```

```
                if Rec[z]==None:
                    d=d+1
            if R[1][z]!=None:
                Rec[z]=1


        if d==len(H)-1:
            trobat=1
        x=x+1

    z=z+1


d=0
trobat=0
z=0
T1=[None for j in range (len(H))]
Rec=[None for j in range (len(H))]
T=[None for j in range (len(H))]
t=[None for j in range(len(H))]
x=0
T[z]=0
while x<len(H):
    if Vsubst[x][0][0]!=None:
        if Vsubst[x][0][1]!=None and Vsubst[x][0][1]>Vsubst[x][0][0]:
            T1[z]=R[1][x]+Vsubst[x][0][1]
            if T1[z]>T[z]:
                T[z]=T1[z]
        else:
            T1[z]=R[1][x]+Vsubst[x][0][0]
            if T1[z]>T[z]:
                T[z]=T1[z]
    x=x+1
t=[None for i in range (len(H))]
t[0]=T[0]
R[2][0]=T[0]
while trobat==0 and d<len(H):
    z=0
    while z<len(H):
        x=0
        while x<len(H):
            if Vsubst[x][z][0]!=None and t[x]!=None and T[z]==None:
                if t[x]>=R[1][x]:
                    T[z]=t[x]+Vsubst[x][z][0]
                else:
                    T[z]=R[1][x]+Vsubst[x][z][0]
                w=0
                while w<len(H):
                    if Vsubst[w][z][0]!=None:
                        if R[1][w]+Vsubst[w][z][0]>T[z]:
                            T[z]=R[1][w]+Vsubst[w][z][0]
                    else:
                        pass
                    w=w+1
```

```
                if t[z]==None or T[z]>t[z]:
                    if T[z]>0:
                        t[z]=T[z]
                        R[2][z]=t[z]
                        if Rec[z]==None:
                            d=d+1
                if R[1][z]!=None:
                    Rec[z]=1

            x=x+1
            if d==len(H)-1:
                trobat=1

        z=z+1

d=0
trobat=0
z=0
T1=[None for j in range (len(H))]
Rec=[None for j in range (len(H))]
T=[None for j in range (len(H))]
t=[None for j in range(len(H))]
x=0
T[z]=0
while x<len(H):
    if Vsubst[x][0][0]!=None:
        if Vsubst[x][0][1]!=None and Vsubst[x][0][1]>Vsubst[x][0][0]:
            T1[z]=R[2][x]+Vsubst[x][0][1]
            if T1[z]>T[z]:
                T[z]=T1[z]
        else:
            T1[z]=R[2][x]+Vsubst[x][0][0]
            if T1[z]>T[z]:
                T[z]=T1[z]
    x=x+1
t=[None for i in range (len(H))]
t[0]=T[0]
R[3][0]=T[0]
while trobat==0 and d<len(H):
    z=0
    while z<len(H):
        x=0
        while x<len(H):
            if Vsubst[x][z][0]!=None and t[x]!=None and T[z]==None:
                if t[x]>=R[2][x]:
                    T[z]=t[x]+Vsubst[x][z][0]
                else:
                    T[z]=R[2][x]+Vsubst[x][z][0]
                w=0
                while w<len(H):
                    if Vsubst[w][z][0]!=None:
                        if R[2][w]+Vsubst[w][z][0]>T[z]:
                            T[z]=R[2][w]+Vsubst[w][z][0]
```

```
                else:
                    pass
                w=w+1


            if t[z]==None or T[z]>t[z]:
                if T[z]>0:
                    t[z]=T[z]
                    R[3][z]=t[z]
                    if Rec[z]==None:
                        d=d+1
                if R[2][z]!=None:
                    Rec[z]=1

            x=x+1
            if d==len(H)-1:
                trobat=1

        z=z+1

d=0
trobat=0
z=0
T1=[None for j in range (len(H))]
Rec=[None for j in range (len(H))]
T=[None for j in range (len(H))]
t=[None for j in range(len(H))]
x=0
T[z]=0
while x<len(H):
    if Vsubst[x][0][0]!=None:
        if Vsubst[x][0][1]!=None and Vsubst[x][0][1]>Vsubst[x][0][0]:
            T1[z]=R[3][x]+Vsubst[x][0][1]
            if T1[z]>T[z]:
                T[z]=T1[z]
        else:
            T1[z]=R[3][x]+Vsubst[x][0][0]
            if T1[z]>T[z]:
                T[z]=T1[z]
    x=x+1
t=[None for i in range (len(H))]
t[0]=T[0]
R[4][0]=T[0]
while trobat==0 and d<len(H):
    z=0
    while z<len(H):
        x=0
        while x<len(H):
            if Vsubst[x][z][0]!=None and t[x]!=None and T[z]==None:
                if t[x]>=R[3][x]:
                    T[z]=t[x]+Vsubst[x][z][0]
                else:
                    T[z]=R[3][x]+Vsubst[x][z][0]
                w=0
```

```
            while w<len(H):
                if Vsubst[w][z][0]!=None:
                    if R[2][w]+Vsubst[w][z][0]>T[z]:
                        T[z]=R[3][w]+Vsubst[w][z][0]
                else:
                    pass
                w=w+1


            if t[z]==None or T[z]>t[z]:
                if T[z]>0:
                    t[z]=T[z]
                    R[4][z]=t[z]
                    if Rec[z]==None:
                        d=d+1
            if R[3][z]!=None:
                Rec[z]=1

        x=x+1
        if d==len(H)-1:
            trobat=1

    z=z+1

d=0
trobat=0
z=0
T1=[None for j in range (len(H))]
Rec=[None for j in range (len(H))]
T=[None for j in range (len(H))]
t=[None for j in range(len(H))]
x=0
T[z]=0
while x<len(H):
    if Vsubst[x][0][0]!=None:
        if Vsubst[x][0][1]!=None and Vsubst[x][0][1]>Vsubst[x][0][0]:
            T1[z]=R[4][x]+Vsubst[x][0][1]
            if T1[z]>T[z]:
                T[z]=T1[z]
        else:
            T1[z]=R[4][x]+Vsubst[x][0][0]
            if T1[z]>T[z]:
                T[z]=T1[z]
    x=x+1
t=[None for i in range (len(H))]
t[0]=T[0]
R[5][0]=T[0]
while trobat==0 and d<len(H):
    z=0
    while z<len(H):
        x=0
        while x<len(H):
            if Vsubst[x][z][0]!=None and t[x]!=None and T[z]==None:
                if t[x]>=R[4][x]:
```

```
                T[z]=t[x]+Vsubst[x][z][0]
            else:
                T[z]=R[4][x]+Vsubst[x][z][0]
            w=0
            while w<len(H):
                if Vsubst[w][z][0]!=None:
                    if R[4][w]+Vsubst[w][z][0]>T[z]:
                        T[z]=R[4][w]+Vsubst[w][z][0]
                else:
                    pass
                w=w+1


            if t[z]==None or T[z]>t[z]:
                if T[z]>0:
                    t[z]=T[z]
                    R[5][z]=t[z]
                    if Rec[z]==None:
                        d=d+1
            if R[4][z]!=None:
                Rec[z]=1

        x=x+1
        if d==len(H)-1:
            trobat=1

    z=z+1
d=0
trobat=0
z=0
T1=[None for j in range (len(H))]
Rec=[None for j in range (len(H))]
T=[None for j in range (len(H))]
t=[None for j in range(len(H))]
x=0
T[z]=0
while x<len(H):
    if Vsubst[x][0][0]!=None:
        if Vsubst[x][0][1]!=None and Vsubst[x][0][1]>Vsubst[x][0][0]:
            T1[z]=R[5][x]+Vsubst[x][0][1]
            if T1[z]>T[z]:
                T[z]=T1[z]
        else:
            T1[z]=R[5][x]+Vsubst[x][0][0]
            if T1[z]>T[z]:
                T[z]=T1[z]
    x=x+1
t=[None for i in range (len(H))]
t[0]=T[0]
R[6][0]=T[0]
while trobat==0 and d<len(H):
    z=0
    while z<len(H):
        x=0
```

```
        while x<len(H):
            if Vsubst[x][z][0]!=None and t[x]!=None and T[z]==None:
                if t[x]>=R[5][x]:
                    T[z]=t[x]+Vsubst[x][z][0]
                else:
                    T[z]=R[5][x]+Vsubst[x][z][0]
                w=0
                while w<len(H):
                    if Vsubst[w][z][0]!=None:
                        if R[5][w]+Vsubst[w][z][0]>T[z]:
                            T[z]=R[5][w]+Vsubst[w][z][0]
                    else:
                        pass
                    w=w+1


                if t[z]==None or T[z]>t[z]:
                    if T[z]>0:
                        t[z]=T[z]
                        R[6][z]=t[z]
                        if Rec[z]==None:
                            d=d+1
                if R[5][z]!=None:
                    Rec[z]=1

            x=x+1
            if d==len(H)-1:
                trobat=1

        z=z+1
d=0
trobat=0
z=0
T1=[None for j in range (len(H))]
Rec=[None for j in range (len(H))]
T=[None for j in range (len(H))]
t=[None for j in range(len(H))]
x=0
T[z]=0
while x<len(H):
    if Vsubst[x][0][0]!=None:
        if Vsubst[x][0][1]!=None and Vsubst[x][0][1]>Vsubst[x][0][0]:
            T1[z]=R[6][x]+Vsubst[x][0][1]
            if T1[z]>T[z]:
                T[z]=T1[z]
        else:
            T1[z]=R[6][x]+Vsubst[x][0][0]
            if T1[z]>T[z]:
                T[z]=T1[z]
    x=x+1
t=[None for i in range (len(H))]
t[0]=T[0]
R[7][0]=T[0]
while trobat==0 and d<len(H):
```

```
    z=0
    while z<len(H):
        x=0
        while x<len(H):
            if Vsubst[x][z][0]!=None and t[x]!=None and T[z]==None:
                if t[x]>=R[6][x]:
                    T[z]=t[x]+Vsubst[x][z][0]
                else:
                    T[z]=R[6][x]+Vsubst[x][z][0]
                w=0
                while w<len(H):
                    if Vsubst[w][z][0]!=None:
                        if R[6][w]+Vsubst[w][z][0]>T[z]:
                            T[z]=R[6][w]+Vsubst[w][z][0]
                    else:
                        pass
                    w=w+1


                if t[z]==None or T[z]>t[z]:
                    if T[z]>0:
                        t[z]=T[z]
                        R[7][z]=t[z]
                        if Rec[z]==None:
                            d=d+1
                if R[6][z]!=None:
                    Rec[z]=1

            x=x+1
            if d==len(H)-1:
                trobat=1

        z=z+1

d=0
trobat=0
z=0
T1=[None for j in range (len(H))]
Rec=[None for j in range (len(H))]
T=[None for j in range (len(H))]
t=[None for j in range(len(H))]
x=0
T[z]=0
while x<len(H):
    if Vsubst[x][0][0]!=None:
        if Vsubst[x][0][1]!=None and Vsubst[x][0][1]>Vsubst[x][0][0]:
            T1[z]=R[7][x]+Vsubst[x][0][1]
            if T1[z]>T[z]:
                T[z]=T1[z]
        else:
            T1[z]=R[7][x]+Vsubst[x][0][0]
            if T1[z]>T[z]:
                T[z]=T1[z]
    x=x+1
```

```
t=[None for i in range (len(H))]
t[0]=T[0]
R[8][0]=T[0]
while trobat==0 and d<len(H):
   z=0
   while z<len(H):
      x=0
      while x<len(H):
         if Vsubst[x][z][0]!=None and t[x]!=None and T[z]==None:
            if t[x]>=R[7][x]:
               T[z]=t[x]+Vsubst[x][z][0]
            else:
               T[z]=R[7][x]+Vsubst[x][z][0]
            w=0
            while w<len(H):
               if Vsubst[w][z][0]!=None:
                  if R[7][w]+Vsubst[w][z][0]>T[z]:
                     T[z]=R[7][w]+Vsubst[w][z][0]
               else:
                  pass
               w=w+1


            if t[z]==None or T[z]>t[z]:
               if T[z]>0:
                  t[z]=T[z]
                  R[8][z]=t[z]
                  if Rec[z]==None:
                     d=d+1
            if R[7][z]!=None:
               Rec[z]=1

         x=x+1
         if d==len(H)-1:
            trobat=1

      z=z+1
d=0
trobat=0
z=0
T1=[None for j in range (len(H))]
Rec=[None for j in range (len(H))]
T=[None for j in range (len(H))]
t=[None for j in range(len(H))]
x=0
T[z]=0
while x<len(H):
   if Vsubst[x][0][0]!=None:
      if Vsubst[x][0][1]!=None and Vsubst[x][0][1]>Vsubst[x][0][0]:
         T1[z]=R[8][x]+Vsubst[x][0][1]
         if T1[z]>T[z]:
            T[z]=T1[z]
      else:
         T1[z]=R[8][x]+Vsubst[x][0][0]
```

```
            if T1[z]>T[z]:
                T[z]=T1[z]
        x=x+1
    t=[None for i in range (len(H))]
    t[0]=T[0]
    R[9][0]=T[0]
    while trobat==0 and d<len(H):
        z=0
        while z<len(H):
            x=0
            while x<len(H):
                if Vsubst[x][z][0]!=None and t[x]!=None and T[z]==None:
                    if t[x]>=R[8][x]:
                        T[z]=t[x]+Vsubst[x][z][0]
                    else:
                        T[z]=R[8][x]+Vsubst[x][z][0]
                    w=0
                    while w<len(H):
                        if Vsubst[w][z][0]!=None:
                            if R[8][w]+Vsubst[w][z][0]>T[z]:
                                T[z]=R[8][w]+Vsubst[w][z][0]
                        else:
                            pass
                        w=w+1


                    if t[z]==None or T[z]>t[z]:
                        if T[z]>0:
                            t[z]=T[z]
                            R[9][z]=t[z]
                            if Rec[z]==None:
                                d=d+1
                    if R[8][z]!=None:
                        Rec[z]=1

                x=x+1
                if d==len(H)-1:
                    trobat=1

            z=z+1


    j=0
    while j<9 and FI==0:
        i=0
        trobat1=0
        while i<len(R[4]) and trobat1==0 and FI==0:
            if R[j][i]==R[j+1][i]:
                i=i+1
            else:
                trobat1=1

        if trobat1==1:
            j=j+1
```

```
            else:
                break

        if trobat1==0:
            ll[g][1]=1
            FI=1

        else:
            OP=0
            op=[None for j in range (9)]
            j=0
            while j<8:
                op[j]=R[j+1][0]-R[j][0]
                i=0
                while i<len(R[j]) and FI==0:
                    if R[j+1][i]-R[j][i]<op[j] or op[j]==0:
                        if R[j+1][i]-R[j][i]!=0:
                            op[j]=R[j+1][i]-R[j][i]
                    else:
                        pass
                    i=i+1
                j=j+1

            OP=op[0]
            i=0
            while i<7:
                if op[i+1]<op[i]:
                    OP=op[i+1]
                i=i+1


            if TC+OP<=ll[g][6] and FI==0 and OP!=0:
                ll[g][2]=TC+OP
            else:
                ll[g][1]=0
                FI=1

        if FI==1:
            pass

    return (ll)


def escollirvertex(ll,g,n,m,ultimaram):

    TC=0
    acabat=0
    i=0

    if g==0:
        g=0
        TC=ll[0][2]
        TCref=TC
        I=0
```

```
        pass


    else:
        while ll[i][1]!=None:
            i=i+1
        longtot=i
        if ll[i][1]==1:
            TCref=ll[i][2]
        else:
            i2=i
            trobatref=0
            while i>0 and trobatref==0:
                if ll[i2][1]==1 and ll[i2][5]==None:
                    TCref=ll[i2][2]
                    trobatref=1
                i2=i2-1
        i=longtot
        while i>0:
            if ll[i][1]==1 and ll[i][5]==None and ll[i][2]<=TCref:
                TC=ll[i][2]
                TCref=TC
                I=i
            i=i-1
    g=I
    if ultimaram==g:
        acabat=1

    return (g,acabat)
```