Projecte Pràctic Salesman
Resolució amb Branch&Bound

Enunciat: Branch & Bound Aplicat al problema del Viatjant de Comerç

- En aquesta entrega s'ha de solucionar el problema del viatjant de comerç implementant tres versions de branch&bound que es diferencien per l'heurística utilitzada
  - Branch&Bound1. Sempre completar el camí més curt.
  - Branch&Bound2. Seleccionar el camí que minimitzi la longitud del camí més una cota mínima fàcil de calcular del camí que resta per fer.
    - Cota mínima: Suma dels camins mínims que porten als vèrtex a visitar pels que encara no hem pasat. Un camí mínim a un vèrtex a visitar V és el mínim del conjunt de camins que portant des de qualsevol vèrtex de la llista de visites al vèrtex V. S'exclou del conjunt ell mateix i tots els camins que **no formaran part de qualsevol solució**. Exemple: Visits=(a,b,c,d,e) Camí fet= a->b. Conjunt de camins a considerar:
      - Per c: {a-c,b-c,d-c} no conté e-c, ja que aquest tram mai formarà part de la solució.
      - Per d: {a-d,b-d,c-d} no conté e-d, ja que aquest tram mai formarà part de la solució.
      - Per e: {b-e, c-e, d-e} no conté a-e, ja que aquest tram mai formarà part de la solució.

Enunciat: Branch & Bound Aplicat al problema del Viatjant de Comerç

- Branch&Bound3. Seleccionar el camí que minimitzi la longitud del camí més una cota mínima difícil de calcular del camí que resta per fer.
  - Cota mínima: Suma dels camins mínims que porten als vèrtex a visitar pels que encara no hem pasat. Un camí mínim a un vèrtex a visitar V és el mínim del conjunt de camins que portant des de qualsevol vèrtex de la llista de visites al vèrtex V. S'exclou del conjunt ell mateix i tots els trams que **no formaran part de la solució considerat el camí ja recorregut**. Exemple: Visits=(a,b,c,d,e) Camí fet= a->b. Conjunt de camins a considerar:
    - Per c: {b-c,d-c}
    - Per d: {b-d,c-d}
    - Per e: {c-e, d-e}
- Per Branch&Bound3 es pot utilitzar qualsevol altre heurística que considereu que pot donar millors resultats. En aquest cas, s'ha d'explicar la heurística en comentaris al fitxer BranchAndBound.cpp

Projecte Pràctic: Entrega Branch&Bound
Entrega en grup

- Entrega:
  - SalesMan V?.zip amb
    - Identificació de l'alumne al fitxer SalesMan.cpp
    - Implementació dels algorismes al fitxer BranchAndBoundcpp.
      - `CTrack SalesmanTrackBranchAndBound1(CGraph& g, CVisits& visits);`
      - `CTrack SalesmanTrackBranchAndBound2(CGraph& g, CVisits& visits);`
      - `CTrack SalesmanTrackBranchAndBound3(CGraph& g, CVisits& visits);`
    - Projecte compilat amb mode release x64
    - Proves dels algorismes a TestSalesMan (veure informe de correcció per saber quines corresponen a cada algorisme).
      - Graf: *.GR
      - Visites: *.VIS
      - Camí solució (una de les solucions possibles): *.TRK (sense greedy)
    - Abans de entregar executar CleanProject.bat

Projecte Pràctic: Entrega Branch&Bound
Entrega en grup

- Data Límit: veure entrega al campus virtual.
- Entrega un sol alumne dels dos que s'identifiquen en la pràctica.
- Nota:
  - Mateix mètode de càlcul de la nota que l’entrega Greedy i Backtracking
  - NotaEntrega=0.8*NotaTests+0.2*NotaTemps+1*(si els temps milloren els del professor) On:
    - $NotaTests = \#TestsOK/\#TestsTotals$
    - $NotaTemps = 10 * (1 - (ta - tr)/(4tr - tr))$ o
    - ta és la suma dels temps de l’alumne de totes les proves d’una entrega o
    - tr és la suma dels temps de referència de totes les proves d’una entrega
    - Els test OK han de executarse en un temp menor o igual al temps de referencia per quatre.
- Recordeu que en aquesta entrega es pot recuperar o millorar la nota de les entregues anteriors.

Branch & Bound Aplicat al problema del Viatjant (Primera versió NO IMPLEMENTAR)

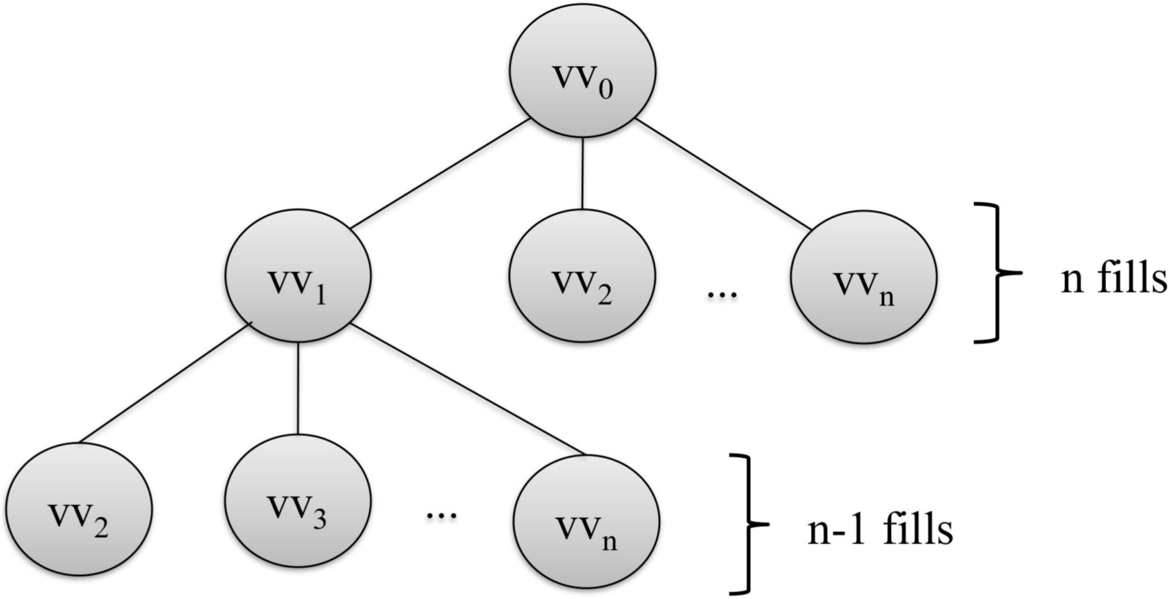
- Solució parcial
  - Un camí pel graf (llista de vèrtexs per on passa).
- Solució arrel de l’arbre de cerca
  - Un camí que només conté el vèrtex origen (primer vèrtex de la llista visits).
- Afegir un element a una solució parcial
  - Afegir al camí de la solució parcial un dels veïns del vèrtex final del camí.
- Heurística simple
  - Seleccionar sempre la solució parcial més curta.

Branch & Bound Aplicat al problema del Viatjant (Segona versió)

- Reduïm l'espai de cerca a l'ordre en que visitarem els vèrtexs de la llista visits (vèrtexs a visitar).
  - Donat un graf G i una llista de vèrtexs a visitar $VV=[vv_0,vv_1\ vv_2, \ldots , vv_n]$
  - Espai de cerca. Les permutacions de la tupla $T=(vv_0,vv_{i1},vv_{i2},vv_{i3},\ldots,vv_{in}, vv_n)$ que comencen per vv_0 i acaben per vv_n.
  - El camí corresponent a la tupla T estarà format per la concatenació dels camins mínims de vv_0 a vv_{i1}, de vv_{i1} a vv_{i2},..., de vv_{in} a vv_n
  - Els camins mínims entre els vèrtexs a visitar es poden calcular utilitzant l'algorisme de Dijkstra

<u>Branch & Bound Aplicat al problema del Viatjant (Segona versió)</u>

- Mida de l'espai de cerca per n vèrtexs a visitar: **(n-2)!**



Heurístiques simples

- De tots els camins guardats a la cua seleccionar
  - El més curt de tots
  - El que minimitzi longitud del camí/vèrtex visitats
  - El que minimitzi longitud del camí més una cota mínima de la longitud del camí que li queda per completar.
    - Cota mínima. Suma les distàncies mínimes per arribar a cada vèrtex no visitat des de qualsevol vèrtex de la llista de visites.
    - Cota mínima. Suma les distàncies mínimes per arribar a cada vèrtex no visitat des de el últim vèrtex visitat o des de vèrtex encara no visitats.

mètodes de poda

- No completar els camins més llargs que el millor trobat fins el moment
- No completar els camins que la seva longitud més la cota mínima per completar-lo és superior que la longitud del camí més curt trobat fins el moment
- Primer buscar un camí per un algorisme greedy i podar els camins amb longitud superior.

Implementació de les heurístiques

Càlculs preliminars a fer

- Per poder calcular les heurístiques s'ha d'emplenar una taula amb tots els camins entre els vèrtexs a visitar:
  - Vèrtex a visitar: [v1,v2,v3,v4,v5,v6]
  - Taula de camins a generar
    - Vi→Vj: Camí del vèrtex i al j de la llista de vèrtexs a visitar
    - No cal emplenar els camins que surten de V6, ja que es el destí final i no sortirem mai d'ell.

Destí

Origen

Camins	0: V1	1: V2	2: V3	3: V4	4: V5	5: V6
0: V1	V1→V1	V1→V2	V1→V3	V1→V4	V1→V5	V1→V6
1: V2	V2→V1	V2→V2	V2→V3	V2→V4	V2→V5	V2→V6
2: V3	V3→V1	V3→V2	V3→V3	V3→V4	V3→V5	V3→V6
3: V4	V4→V1	V4→V2	V4→V3	V4→V4	V4→V5	V4→V64:
4: V5	V5→V1	V5→V2	V5→V3	V5→V4	V5→V5	V5→V6
5: V6						

Que és un camí?

- Un camí es representa pels trams que el composen. La millor forma de representar-lo és amb la llista d'índex dels vèrtexs a visitar que formen els trams
  - Trams: V1→V2, V2→V4, V4→V5, V5→V3, V3→V6
  - Llista d'índexs: [0, 2, 3, 4, 2, 5]
- Calcular la longitud del camí
  - Camins[0,2]+Camins[2,3]+Camins[3,4]+Camins[4,2]+Camins[2,5]
- Com es calculen les heurístiques o cotes:
  - Heurística de cada camí: longitud del camí + estimació del que falta
  - Cota inferior: longitud del camí + cota inferior del camí que falta
  - Cota superior: longitud del camí + cota superior del camí que falta

Heurística per Branch&Bound1

- Les solucions a la cua amb prioritat s’ordenen segons la longitud del camí que tenen construït del més curt al més llarg.
  - Per tant, s'escollirà com solució parcial més prometedora el camí més curt.
  - Aquesta heurística farà un recorregut per l’arbre de cerca amb amplada prioritària.

Heurística per Branch&Bound2

- Les solucions a la cua amb prioritat s'ordenen segons la longitud del camí més una estimació estàtica del que queda per recorre.
  - Tindrem una cota inferior i una superior per aquesta estimació que calcularem considerant nomes els vèrtexs a visitar pels que no hem passat.
- Exemple:
  - Llista de vèrtexs a visitar [V1, V2, V3, V4, V5 ,V6]
  - Camí: V1→V2, V2→V4: [0,1,3]
  - Falta per visitar els vèrtex V3, V5 i V6. Per tant hem de fer una estimació del que pot costar arribar a aquets vèrtexs. Considerarem els camins d'anar de qualsevol vèrtexs de la llista de visites fins aquest encara no visitats:
    - Camins per anar a V3 CV3= { V1→V3, V2→V3, V4→V3, V5→V3}
    - Camins per anar a V5 CV5= { V1→V5, V2→V5, V3→V5, V4→V5}
    - Camins per anar a V6 CV6= { V1→V6, V2→V6, V3→V6, V5→V6}
  - Cota inferior del camí que hem porti a V3, V5 i V6: min(CV3)+min(CV5)+min(CV6)
  - Cota superior del camí que hem porti a V3, V5 i V6: max(CV3)+max(CV5)+max(CV6)

Heurística per Branch&Bound2

- Detalls de com fer el càlcul de les cotes:
  - Els conjunts CVi son els camins que hi ha a la columa i de la taula de camins treguent el camí que va de CVi a CVi. Per tant, els podem calcular a la inicialització del algorisme fen els mínims i màxims de les columnes.
  - Cami V1→V2→V4:
    - CV3= { V1→V3, V2→V3, V4→V3, V5→V3}
    - CV5= { V1→V5, V2→V5, V3→V5, V4→V5}
    - CV6= { V1→V6, V2→V6, V3→V6, V5→V6}

Camins	0: V1	1: V2	2: V3	3: V4	4: V5	5: V6
0: V1	V1→V1	V1→V2	V1→V3	V1→V4	V1→V5	V1→V6
1: V2	V2→V1	V2→V2	V2→V3	V2→V4	V2→V5	V2→V6
2: V3	V3→V1	V3→V2	V3→V3	V3→V4	V3→V5	V3→V6
3: V4	V4→V1	V4→V2	V4→V3	V4→V4	V4→V5	V4→V6
4: V5	V5→V1	V5→V2	V5→V3	V5→V4	V5→V5	V5→V6
5: V6						

- Cotes del camí buit:
  - Cota superior Suma de max(CVi)
  - Cota inferior Suma de min(CVi)
- Afegeixo al camí [V1→...→Vi] el vèrtex a visitar Vj: Les seves cotes seran:
  - Cota superior [V1→...→Vi]-max(CVj) + longitud de Vi→Vj
  - Cota inferior [V1→...→Vi]-min(CVj) + longitud de Vi→Vj
- El càlcul de les cotes suposa fer una resta i una suma a les cotes anteriors. Restar les cotes d'arribar al vèrtex Vj i sumar la longitud del camí que va de Vi a Vj.

Heurística per Branch&Bound3

- Podeu utilitzar l'heurística que volgué. Nomes proposo una per si la voleu utilitzar.
- Les cotes del camí que falta per recorre en la heurística 2 es poden millorar considerant nomes els camis que realment falten per recorre.
- Exemple:
  - Llista de vèrtexs a visitar [V1, V2, V3, V4, V5 ,V6]
  - Camí: V1→V2, V2→V4: [0,1,3]
  - Falta per visitar els vèrtex V3, V5 i V6. Considerarem els camins que realment es poden seleccionar per anar als vèrtexs V3, V5 i V6: poden sortir del vèrtex final del camí V4 o de qualsevol vèrtex no visitat, sense incloure el final (en vermell els camins considerats per l'heurística 2 que no considera la 3):
    - Camins per anar a V3 CV3'= { V1→V3, V2→V3, V4→V3, V5→V3}
    - Camins per anar a V5 CV5'= { V1→V5, V2→V5, V3→V5, V4→V5}
    - Camins per anar a V6 CV6'= { V1→V6, V2→V6, V3→V6, V5→V6}
  - Cota inferior del camí que hem porti a V3, V5 i V6: min(CV3')+min(CV5')+min(CV6')
  - Cota superior del camí que hem porti a V3, V5 i V6: max(CV3')+max(CV5')+max(CV6')

Heurística per Branch&Bound3

- Detalls de com fer el càlcul de les cotes:
  - Els conjunts CVi’ son alguns dels camins que hi ha a la columna i de la taula de camins, per tant els màxims i mínims es tenen que tornar a calcular de nou per cada camí.
- Exemple camí V1→V2 es consideren els camins en verd:

Camins	0: V1	1: V2	2: V3	3: V4	4: V5	5: V6
0: V1	V1→V1	V1→V2	V1→V3	V1→V4	V1→V5	V1→V6
1: V2	V2→V1	V2→V2	V2→V3	V2→V4	V2→V5	V2→V6
2: V3	V3→V1	V3→V2	V3→V3	V3→V4	V3→V5	V3→V6
3: V4	V4→V1	V4→V2	V4→V3	V4→V4	V4→V5	V4→V6
4: V5	V5→V1	V5→V2	V5→V3	V5→V4	V5→V5	V5→V6
5: V6						

No es pot anar directament al vèrtex final

- Exemple camí V1→V2→V4 es consideren els camins en verd:

Camins	0: V1	1: V2	2: V3	3: V4	4: V5	5: V6
0: V1	V1→V1	V1→V2	V1→V3	V1→V4	V1→V5	V1→V6
1: V2	V2→V1	V2→V2	V2→V3	V2→V4	V2→V5	V2→V6
2: V3	V3→V1	V3→V2	V3→V3	V3→V4	V3→V5	V3→V6
3: V4	V4→V1	V4→V2	V4→V3	V4→V4	V4→V5	V4→V6
4: V5	V5→V1	V5→V2	V5→V3	V5→V4	V5→V5	V5→V6
5: V6						

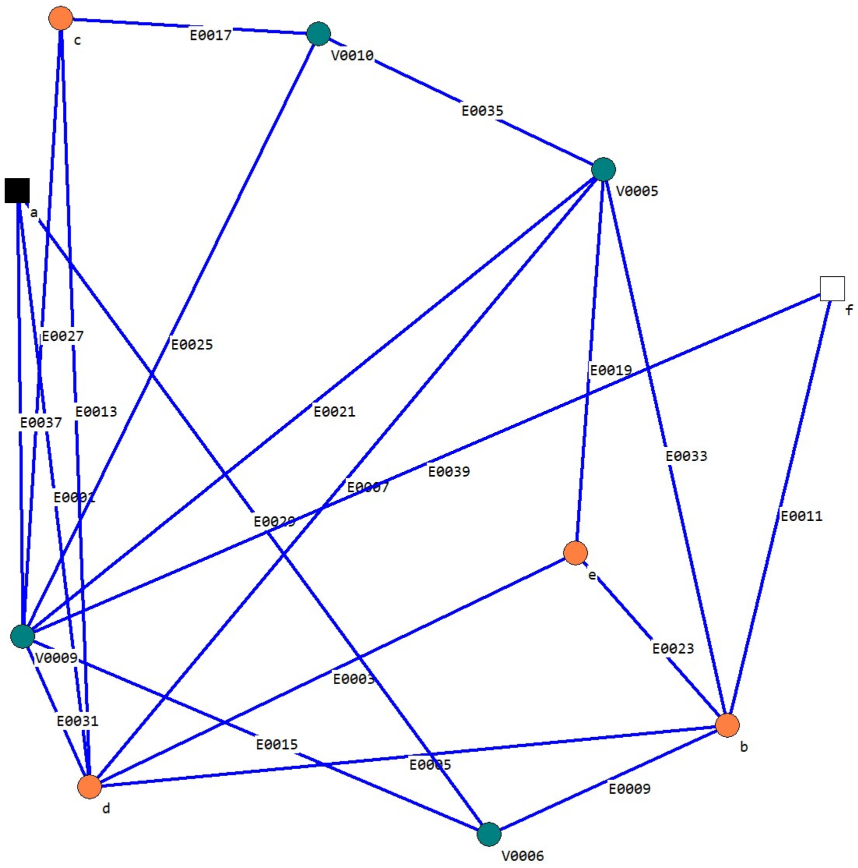
No es pot anar directament al vèrtex final

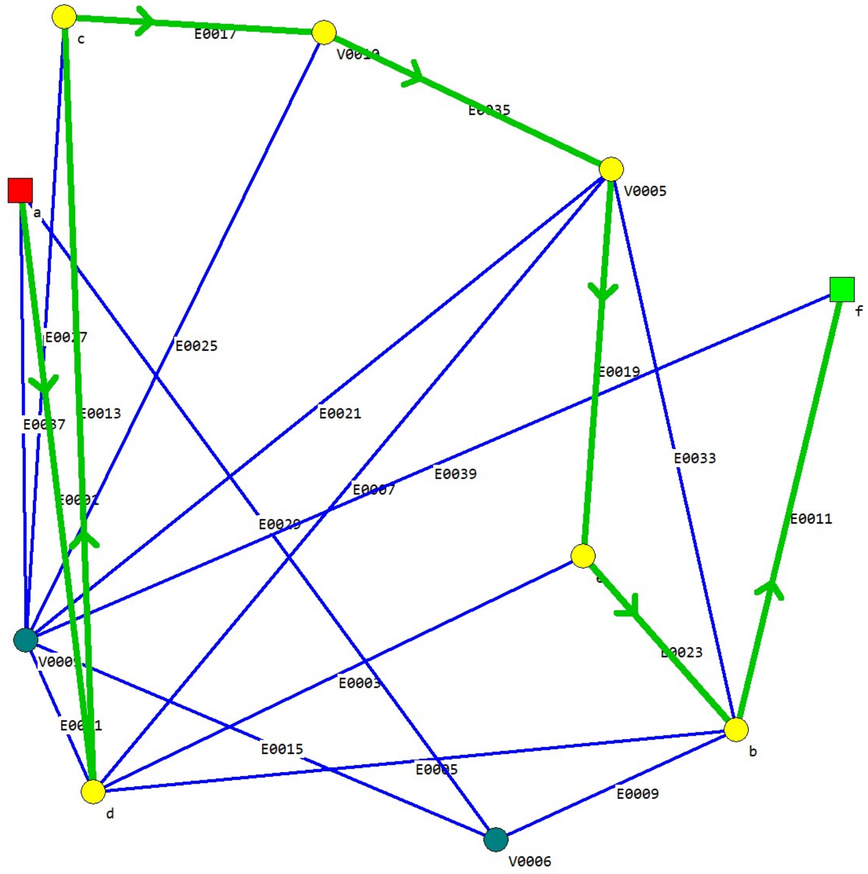
Cotes i Precisió dels càlculs amb double

- Una cosa que s'ha de tenir en compte a podar una solució per cota inferior més alta que la cota superior es el següent:
  - La precisió dels càlculs amb doubles afecta a les sumes:
    - a=1; b=1e-16;
    - Curiosament es compleig que a+b+b!=b+b+a
    - Com el calcul de les cotes no fa les sumes en el mateix ordre que el calcul de la longitud dels camins, es pot donar el cas que la cota inferior de la solució óptima sigui lleugerament superior a la cota superior que utilitcem per podar. Per tant, es interesant sumar un valor molt petit (1e-5) a la cota superior per evitar problemes amb la precisió dels calculs que portin a podar el cami óptim.

Exemple d’execució per l’heurística 3

Graf on cerca el camí del viatjant de comerç





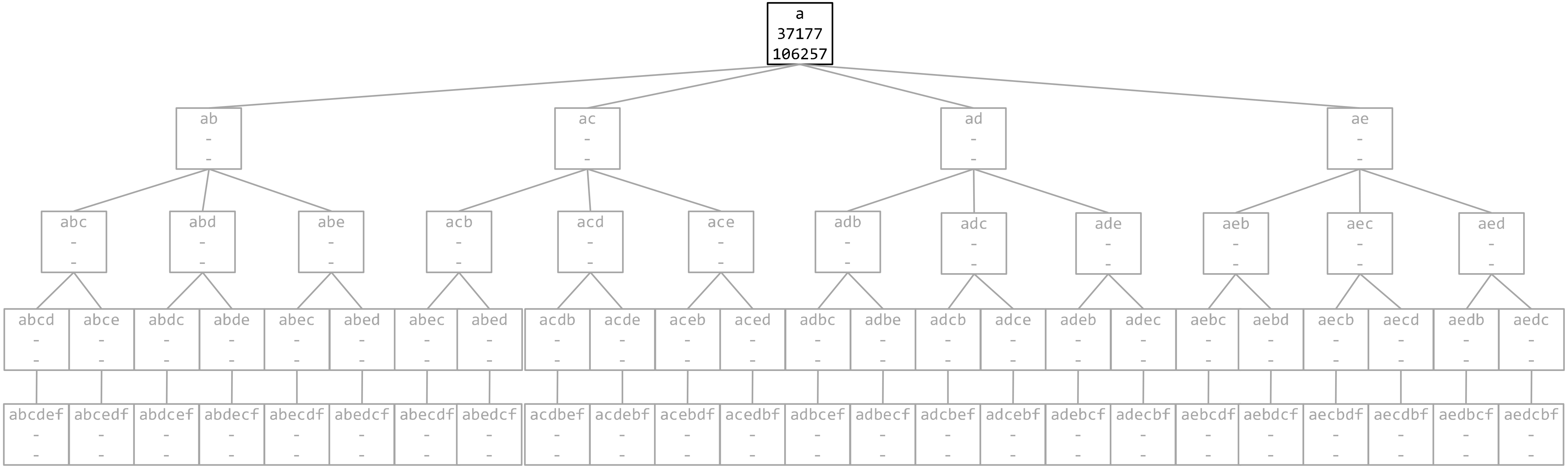
Taula de camins entre visites

- Taula de camins

	a	b	c	d	e	f
a	0	17786.4	17877.6	10069.3	19116.2	22283.5
b	17786.4	0	19187.7	10755.1	3848.92	7533.04
c	17877.6	19187.7	0	12899.4	16090	25197.7
d	10069.3	10755.1	12899.4	0	9046.9	17557.8
e	19116.2	3848.92	16090	9046.9	0	11382
f	22283.5	7533.04	25197.7	17557.8	11382	0

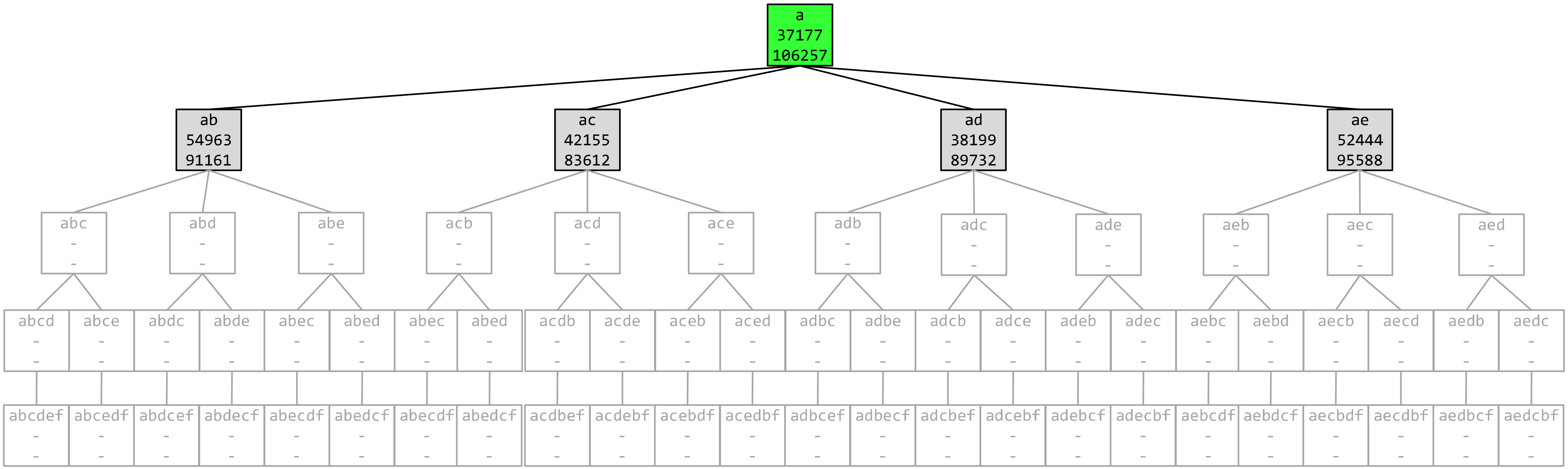
Arbre de camins

- Cota superior global: 106257
- Cua: a



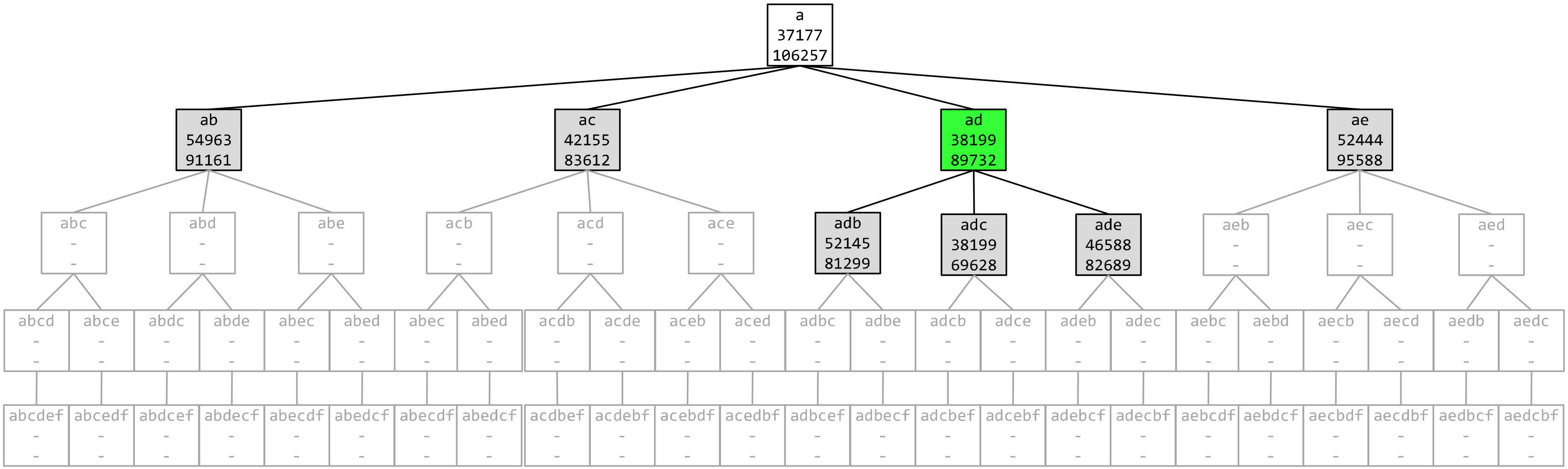
Arbre de camins

- Cota superior global: 106257
- Cua: a
- Ramifica **a**: a→b, a→c, a→d, a→e



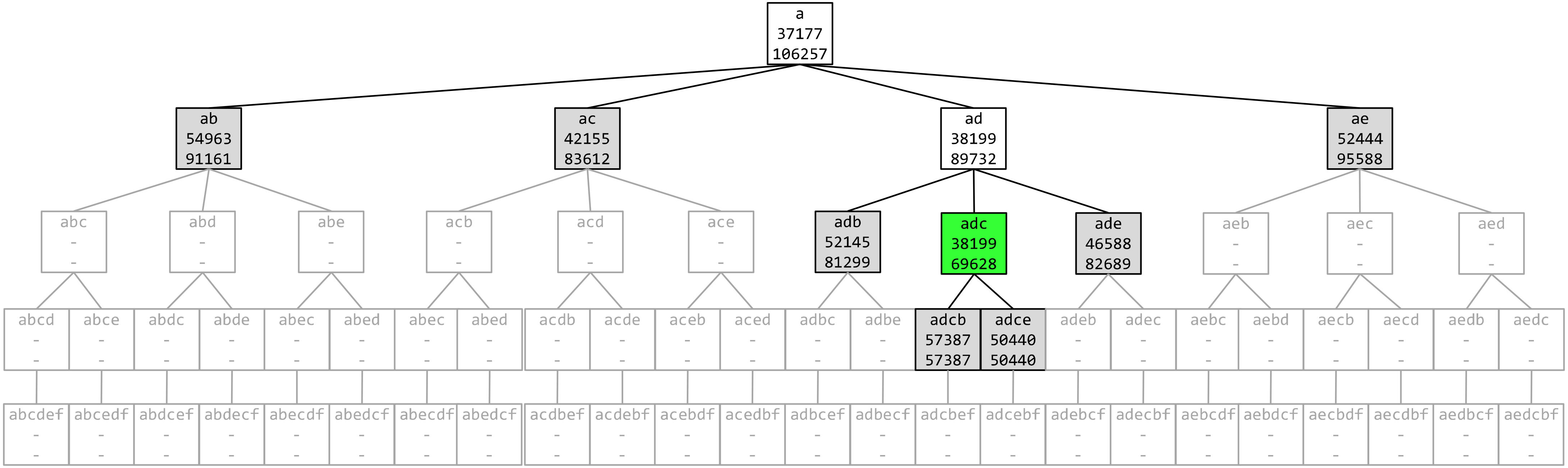
Arbre de camins

- Cota superior global: 83612.5
- Cua: a→d, a→c, a→e, a→b
- Ramifica **a→d**: a→d→c, a→d→b, a→d→e



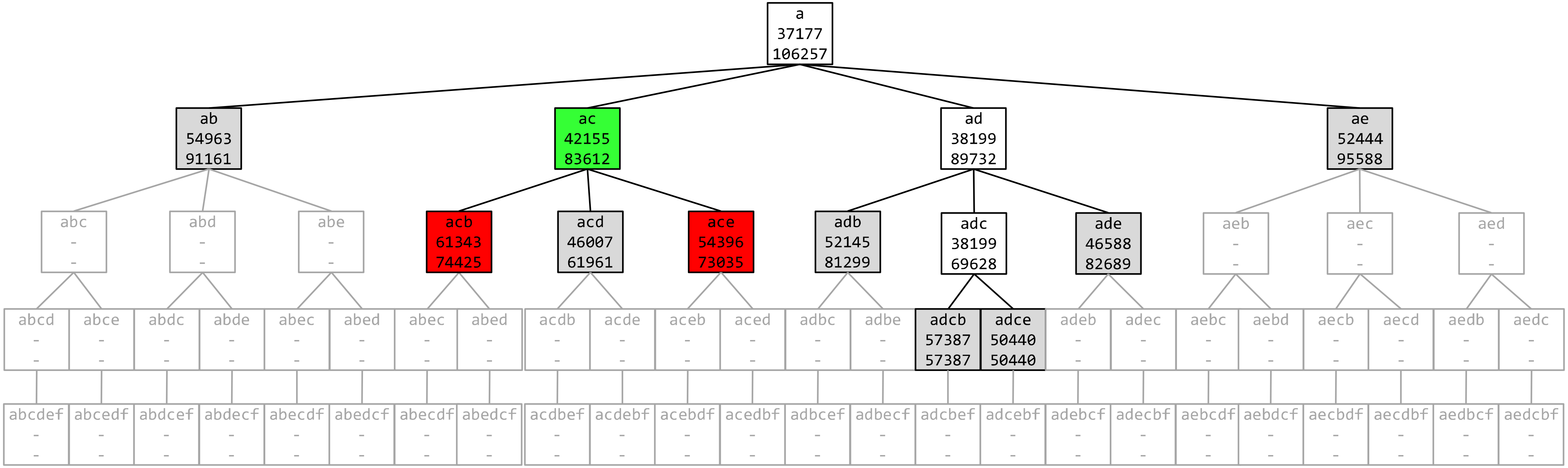
Arbre de camins

- Cota superior global: 69628.4
- Cua: a→d→c, a→c, a→d→e, a→d→b, a→e, a→b
- Ramifica **a→d→c**: a→d→c→b, a→d→c→e



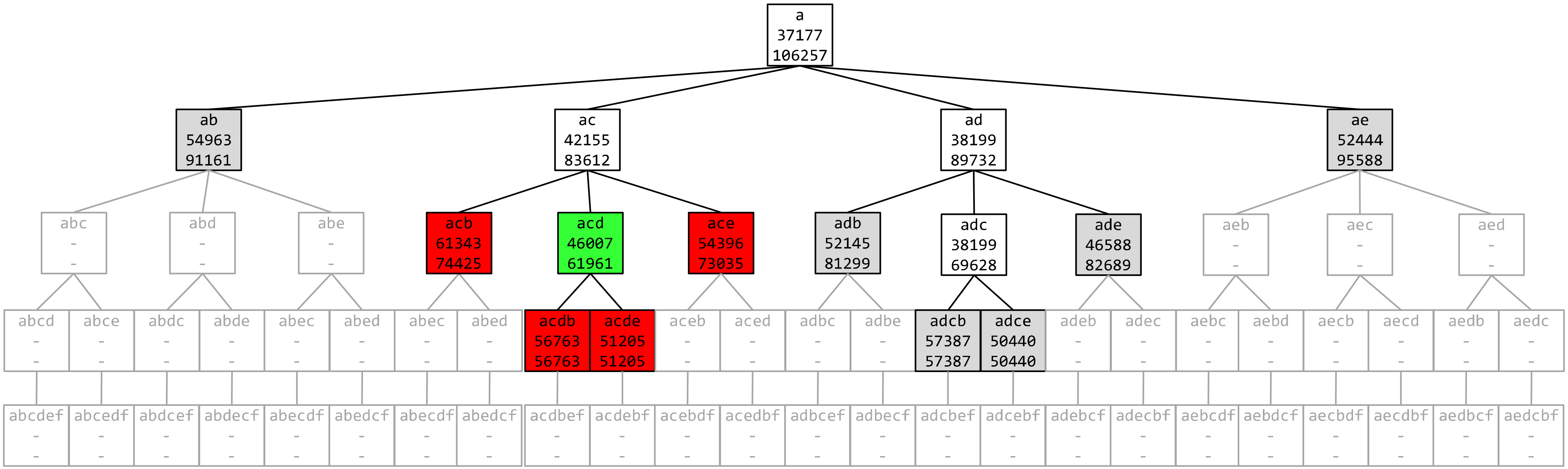
Arbre de camins

- Cota superior global: 50440.7
- Cua: a→c, a→d→e, a→d→c→e, a→d→b, a→e, a→b, a→d→c→b
- Ramifica **a→c**: a→c→b, a→c→d, a→c→e



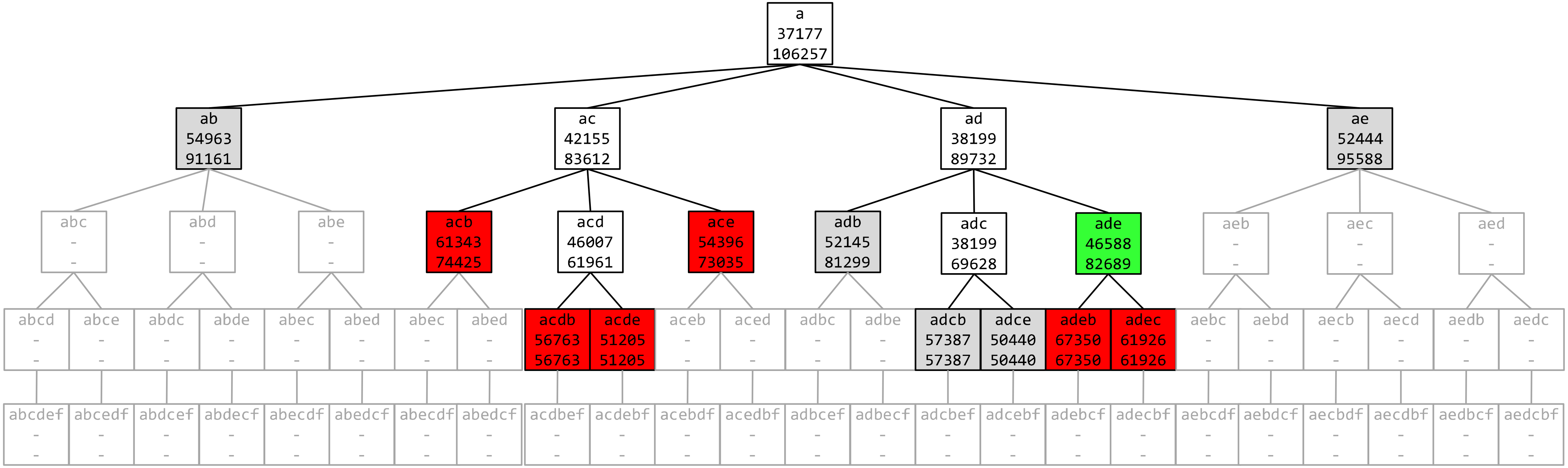
Arbre de camins

- Cota superior global: 50440.7
- Cua: a→c→d, a→d→e, a→d→c→e, a→d→b, a→e, a→b, a→d→c→b
- Ramifica **a→c→d**: a→c→d→b, a→c→d→e



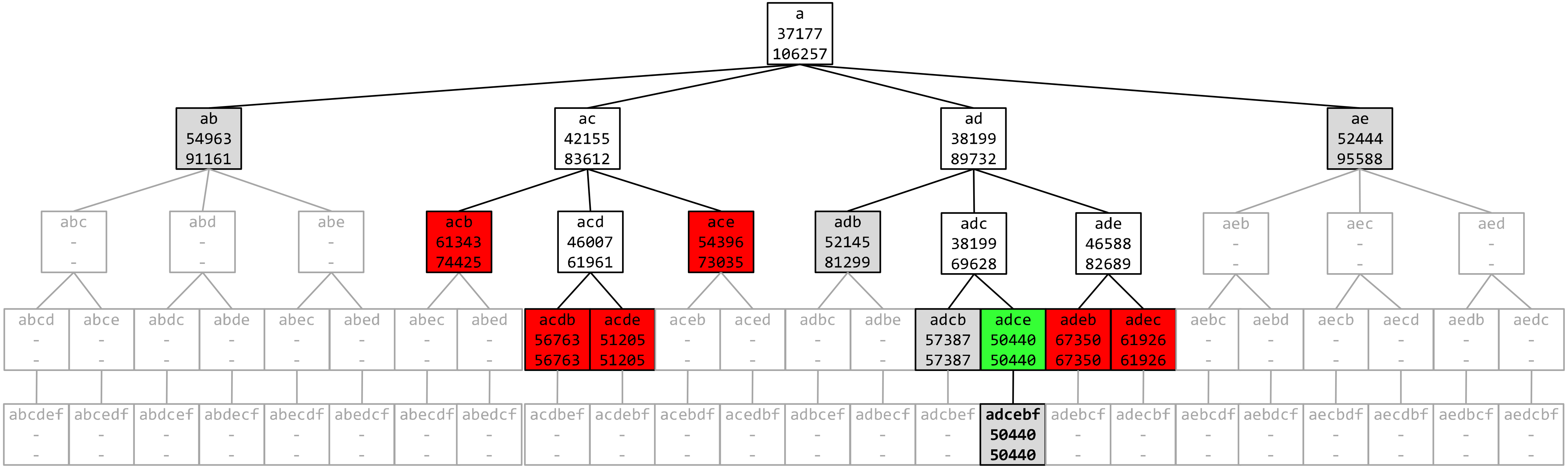
Arbre de camins

- Cota superior global: 50440.7
- Cua: a→d→e, a→d→c→e, a→d→b, a→e, a→b, a→d→c→b
- Ramifica **a→d→e**: a→d→e→b, a→d→e→c



Arbre de camins

- Cota superior global: 50440.7
- Cua: a→d→c→e, a→d→b, a→e, a→b, a→d→c→b
- Ramifica **a→d→c→e**: a→d→c→e→b→f
- Solució més curta trobada fins el moment: a→d→c→e→b→f



Arbre de camins

- Cota superior global: 50440.7
- Cua: a→d→c→e→b→f, a→d→b, a→e, a→b, a→d→c→b
- Solució final: **a→d→c→e→b→f**
- Nodes totals: 65. Nodes Generats: 18 (27.6%). Nodes no generats:46 (70.7%)