

The left side of the slide features a decorative design consisting of several vertical bars of varying heights and widths in shades of light blue and grey. Below these bars, there are five solid teal circles of different sizes, arranged in a cluster that resembles a thought bubble or a group of data points.

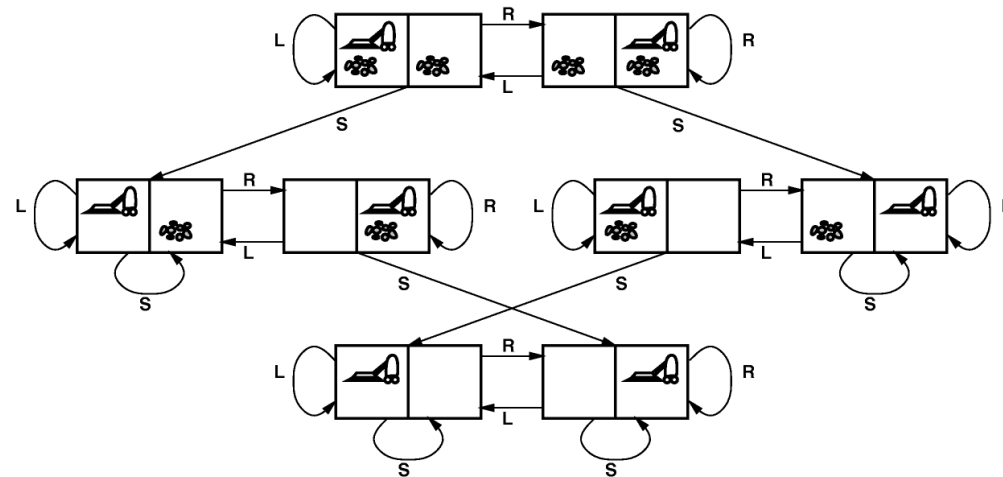
UMETNA INTELIGENCA

neinformirano in hevristično preiskovanje

PROSTOR STANJ

Prostor stanj je:

- formalizem za predstavljanje problemov
- v obliki grafa, pri čemer
 - vozlišča ustrezajo problemskim situacijam
 - povezave ustrezajo dovoljenim akcijam



PREDSTAVITEV PROBLEMA

Problem je definiran s:

- prostorom stanj
- začetnim stanjem
- končnim stanjem (lahko več)

Reševanje problema zahteva preiskovanje grafa:

- rešitev problema je pot od začetnega do končnega stanja
- optimizacijske probleme predstavimo tako, da povezavam v grafu dodamo cene
- cena rešitve je vsota vseh povezav vzdolž rešitvene poti

NEINFORMIRANO PREISKOVANJE

Osnovni strategiji za sistematično preiskovanje prostora stanj:

- iskanje v globino
 - med alternativami izbere tisto, ki je najdlje od začetnega stanja
 - najbolj prilega rekurzivnemu stilu programiranja
 - nevarnost zankanja
 - ni nujno, da najprej najde najkrajšo pot
 - časovna zahtevnost reda $O(b^m)$, prostorska zahtevnost reda $O(bm)$
- iskanje v širino
 - med alternativami izbere tisto najbližjo začetnemu stanju
 - vedno najprej najde najkrajšo pot
 - moramo voditi množico poti-kandidatov
 - časovna zahtevnost reda $O(b^d)$, prostorska zahtevnost reda $O(b^d)$

b – faktor vejanja grafa

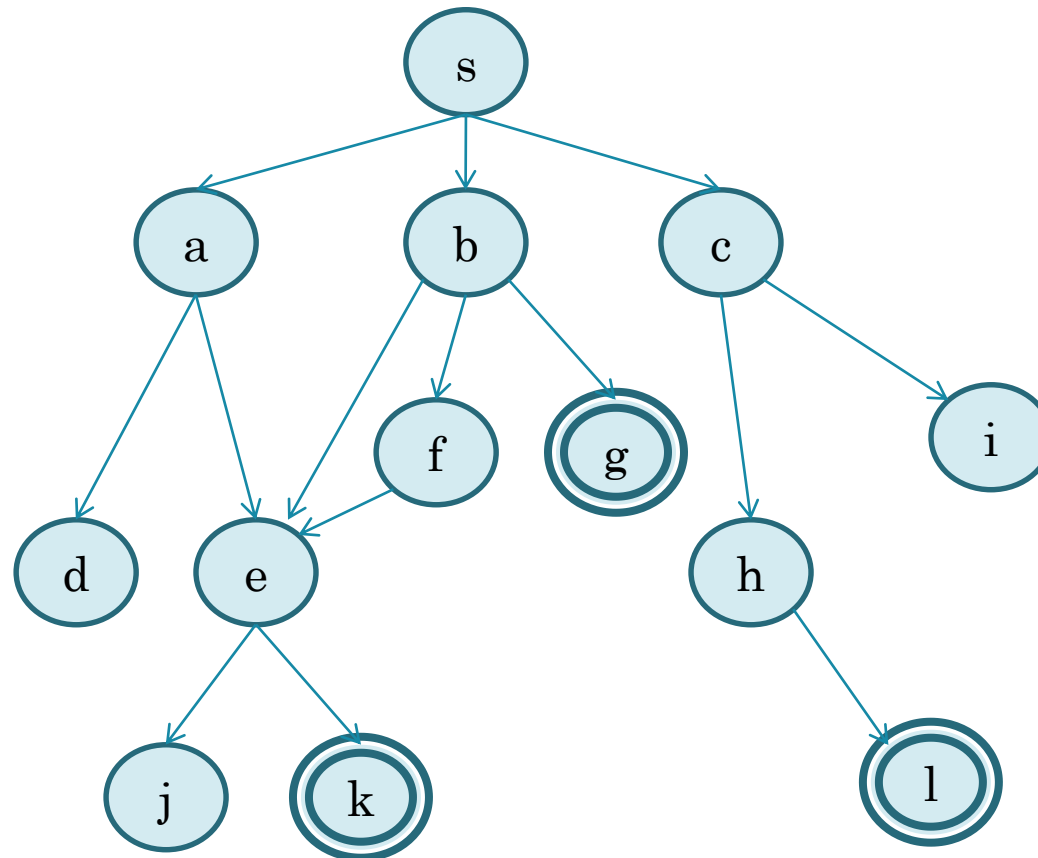
d – globina najbližjega končnega stanja

m – maksimalna globina prostora stanj

PRIMER - ISKANJE V GLOBINO (1/10)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
F	F	F	F	F	F	F	F	F	F	F	F	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	-	-	-	-	-	-	-	-	-	-	-	-



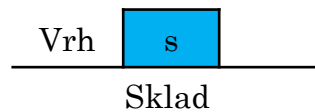
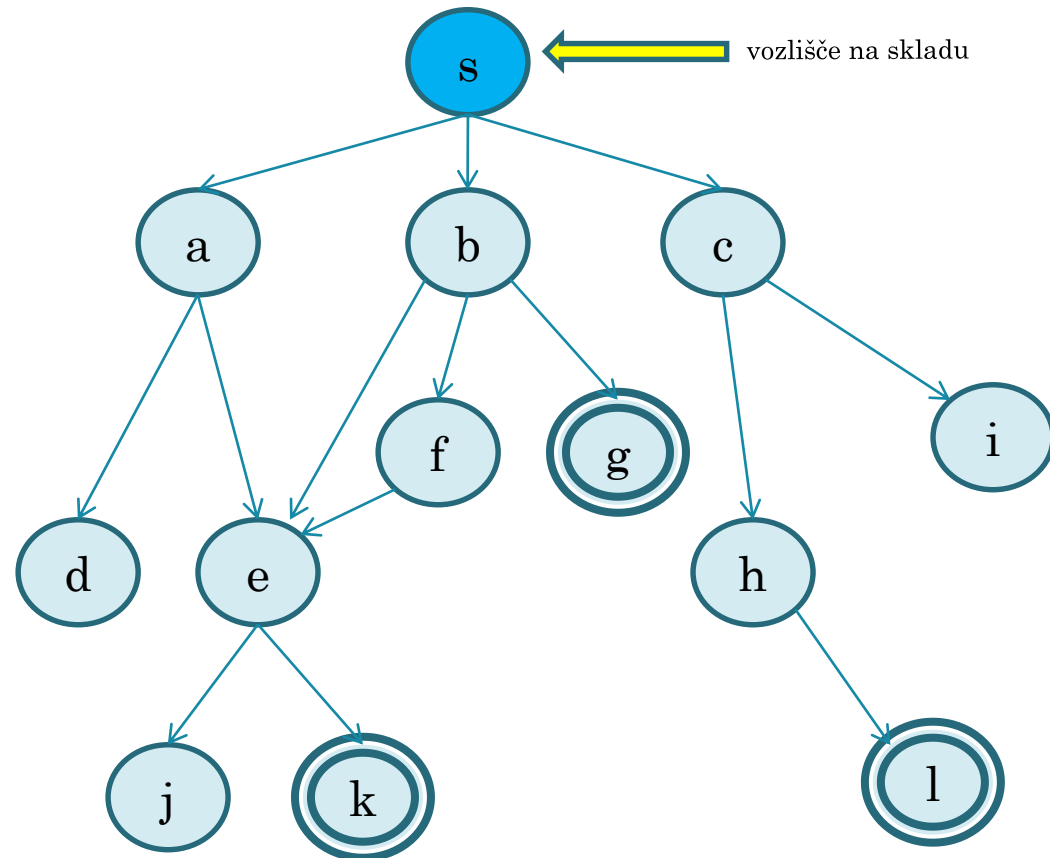
Vrh -

Sklad

PRIMER - ISKANJE V GLOBINO (2/10)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	F	F	F	F	F	F	F	F	F	F	F	F

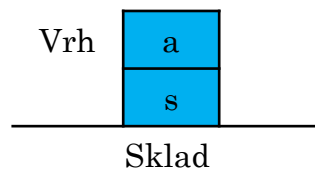
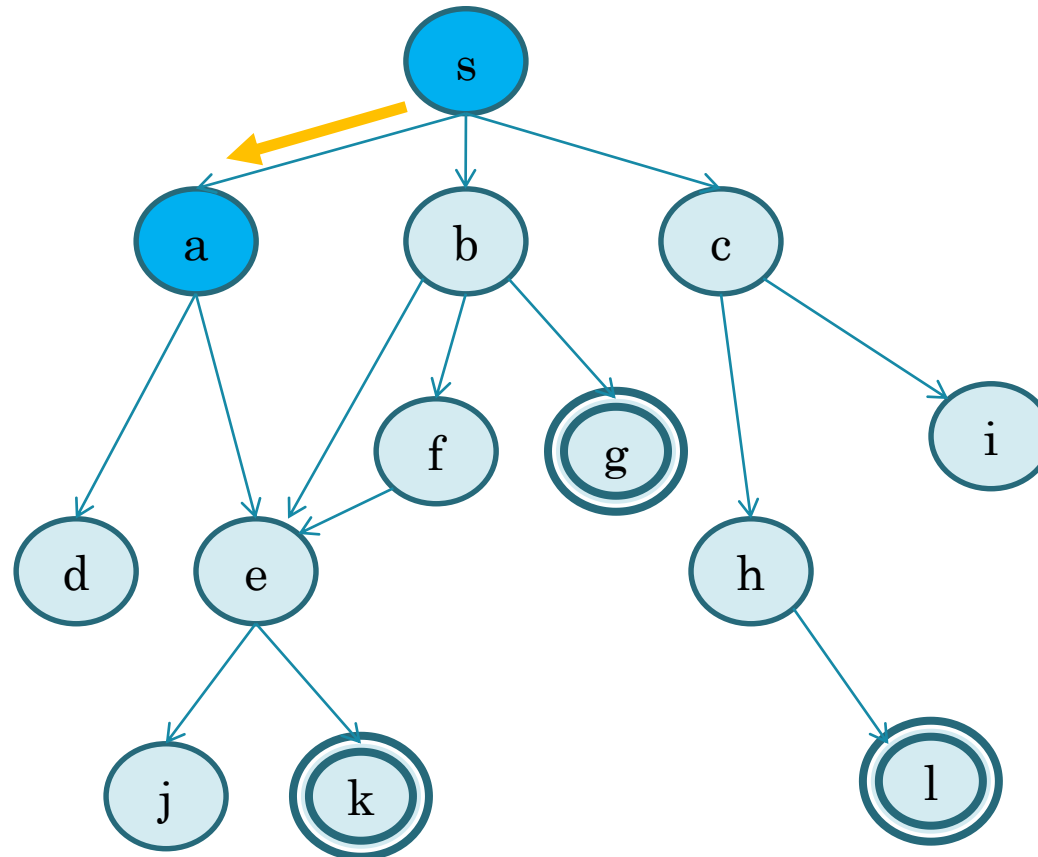
Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	-	-	-	-	-	-	-	-	-	-	-	-



PRIMER - ISKANJE V GLOBINO (3/10)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	F	F	F	F	F	F	F	F	F	F	F

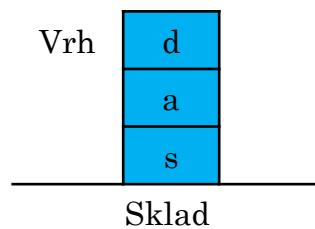
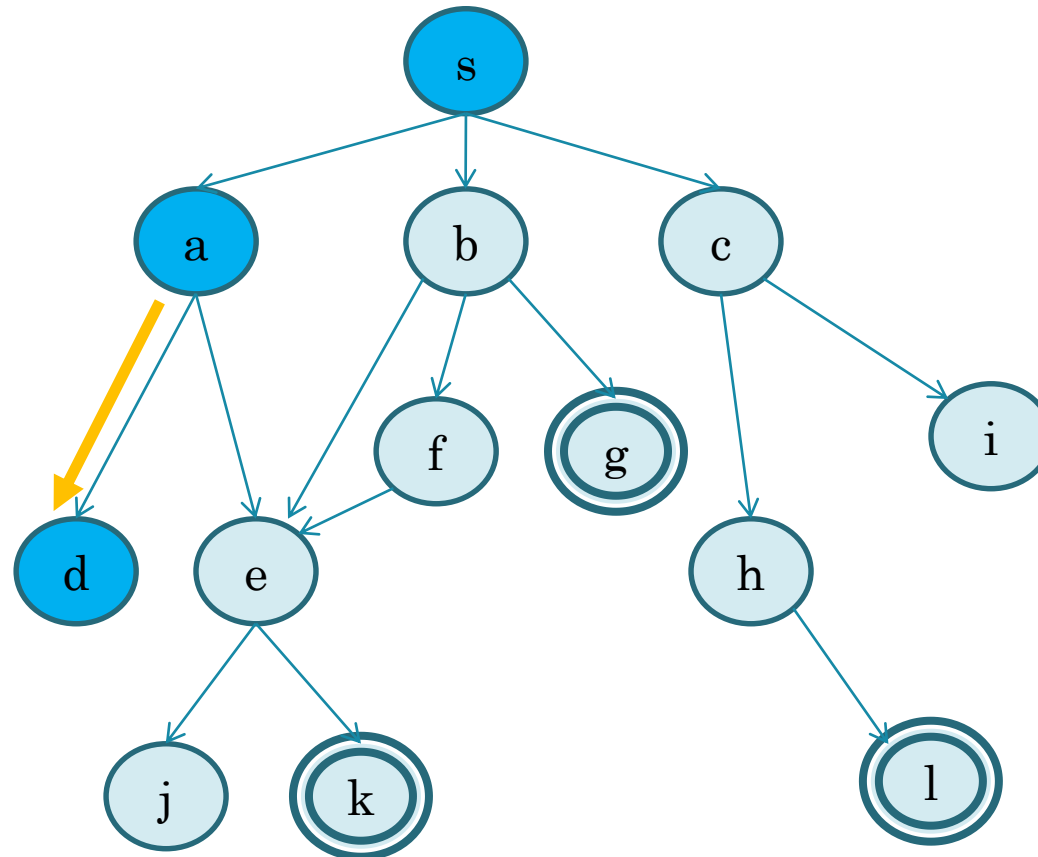
Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	-	-	-	-	-	-	-	-	-	-	-



PRIMER - ISKANJE V GLOBINO (4/10)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	F	F	T	F	F	F	F	F	F	F	F

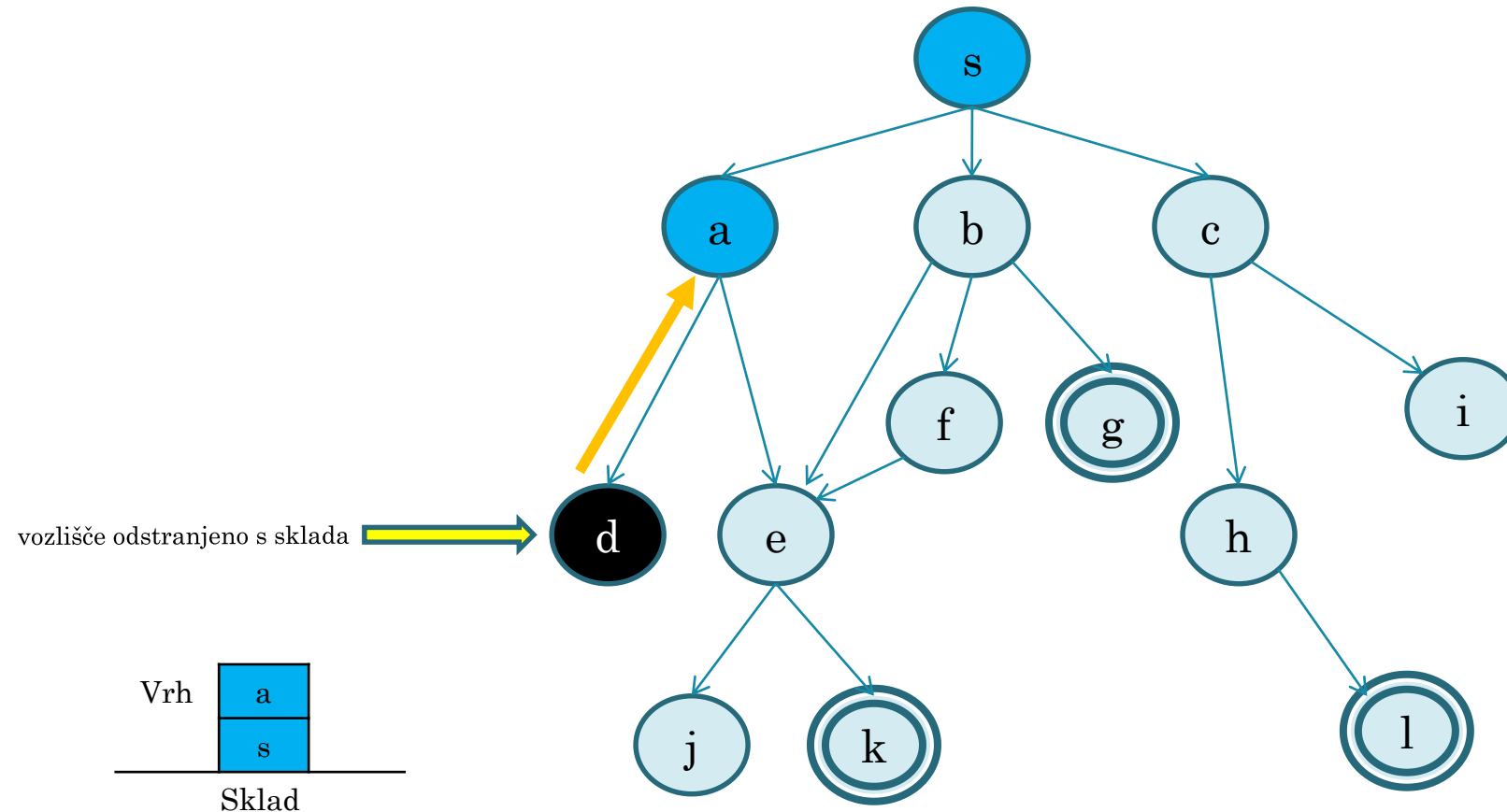
Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	-	-	a	-	-	-	-	-	-	-	-



PRIMER - ISKANJE V GLOBINO (5/10)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	F	F	T	F	F	F	F	F	F	F	F

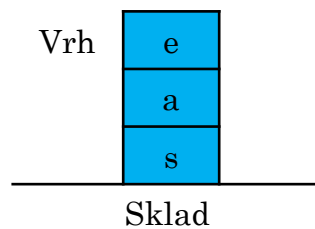
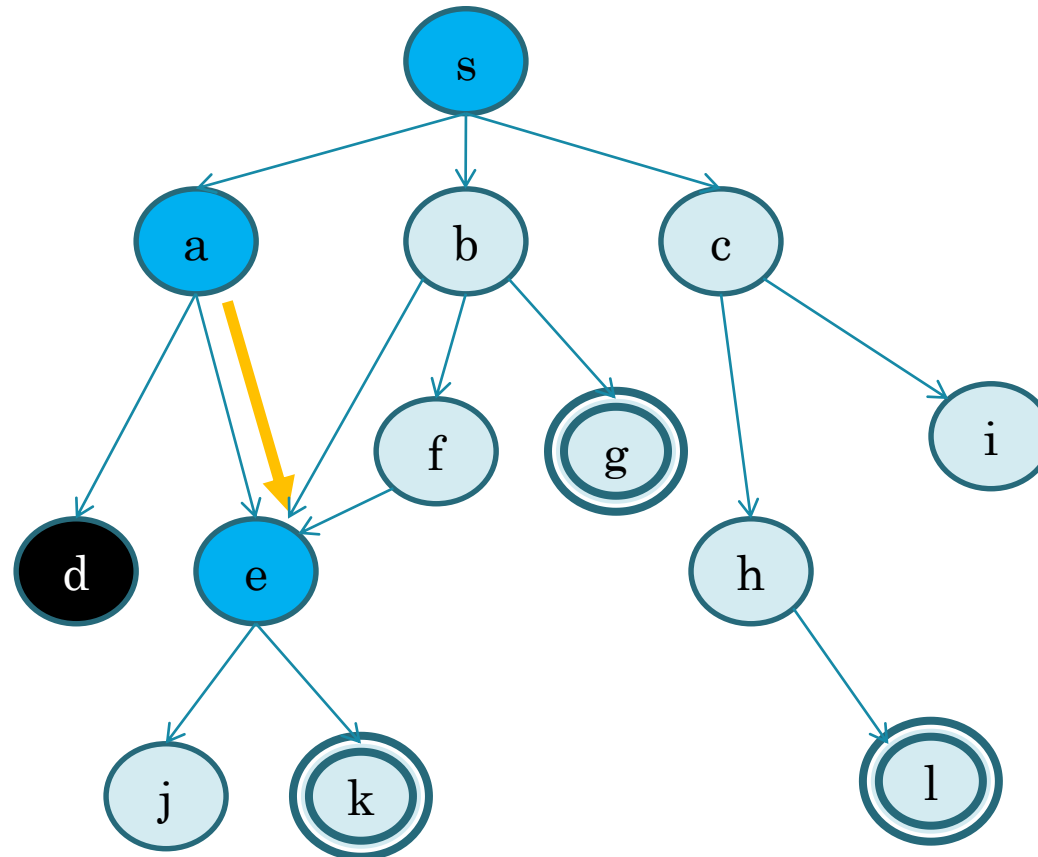
Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	-	-	a	-	-	-	-	-	-	-	-



PRIMER - ISKANJE V GLOBINO (6/10)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	F	F	T	T	F	F	F	F	F	F	F

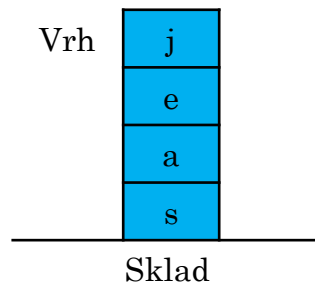
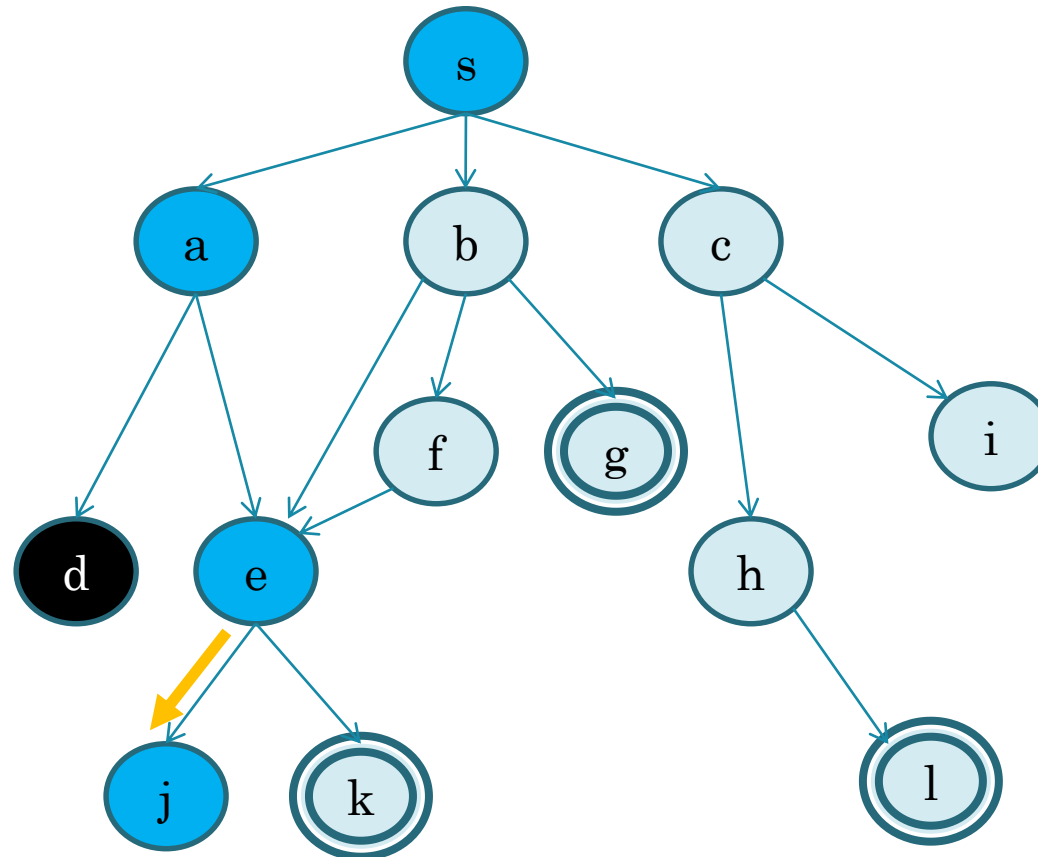
Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	-	-	a	a	-	-	-	-	-	-	-



PRIMER - ISKANJE V GLOBINO (7/10)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	F	F	T	T	F	F	F	F	T	F	F

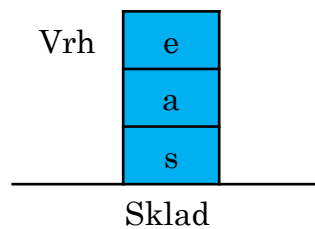
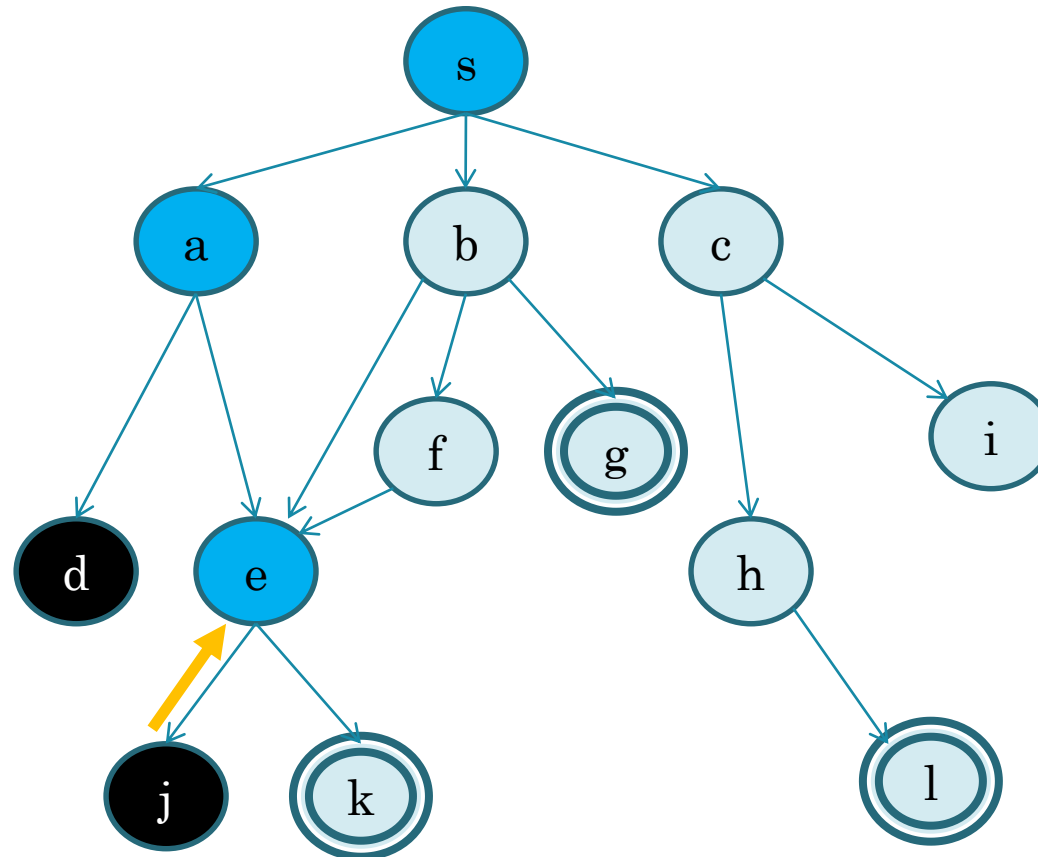
Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	-	-	a	a	-	-	-	-	e	-	-



PRIMER - ISKANJE V GLOBINO (8/10)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	F	F	T	T	F	F	F	F	T	F	F

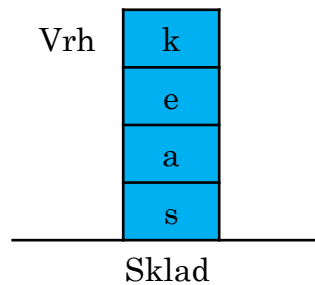
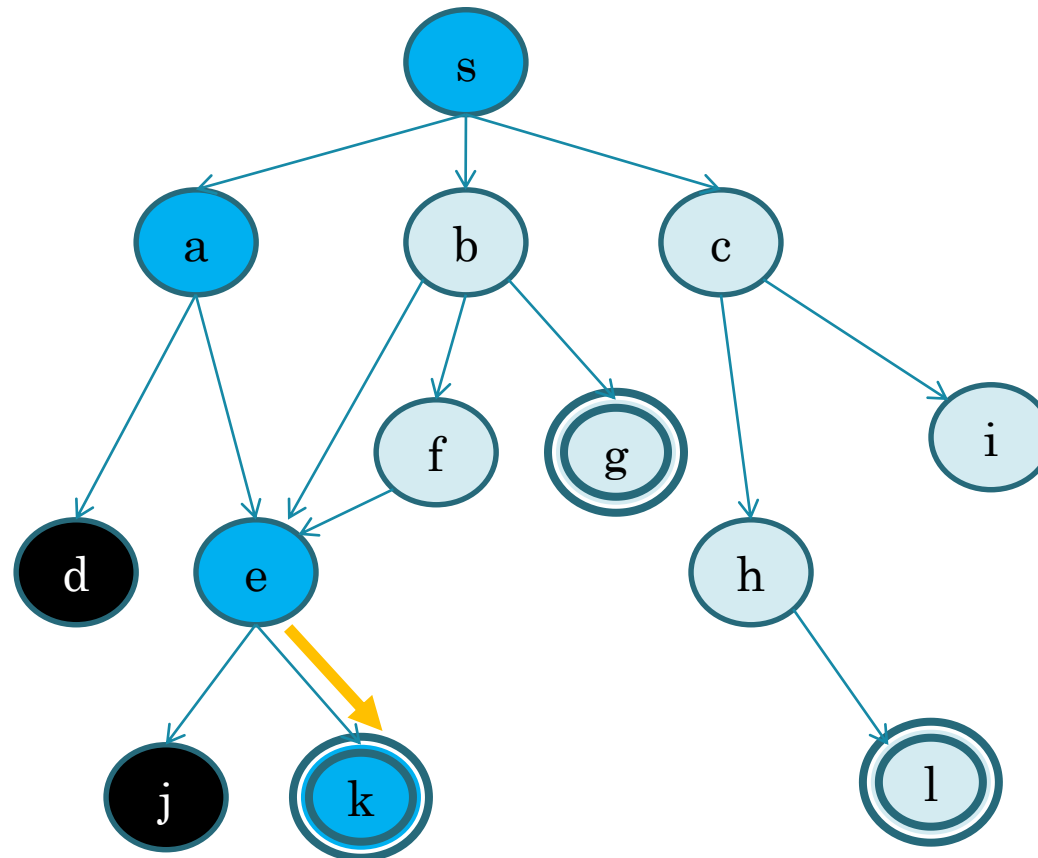
Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	-	-	a	a	-	-	-	-	e	-	-



PRIMER - ISKANJE V GLOBINO (9/10)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	F	F	T	T	F	F	F	F	T	T	F

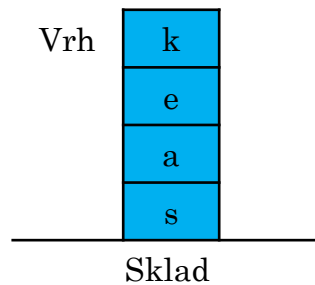
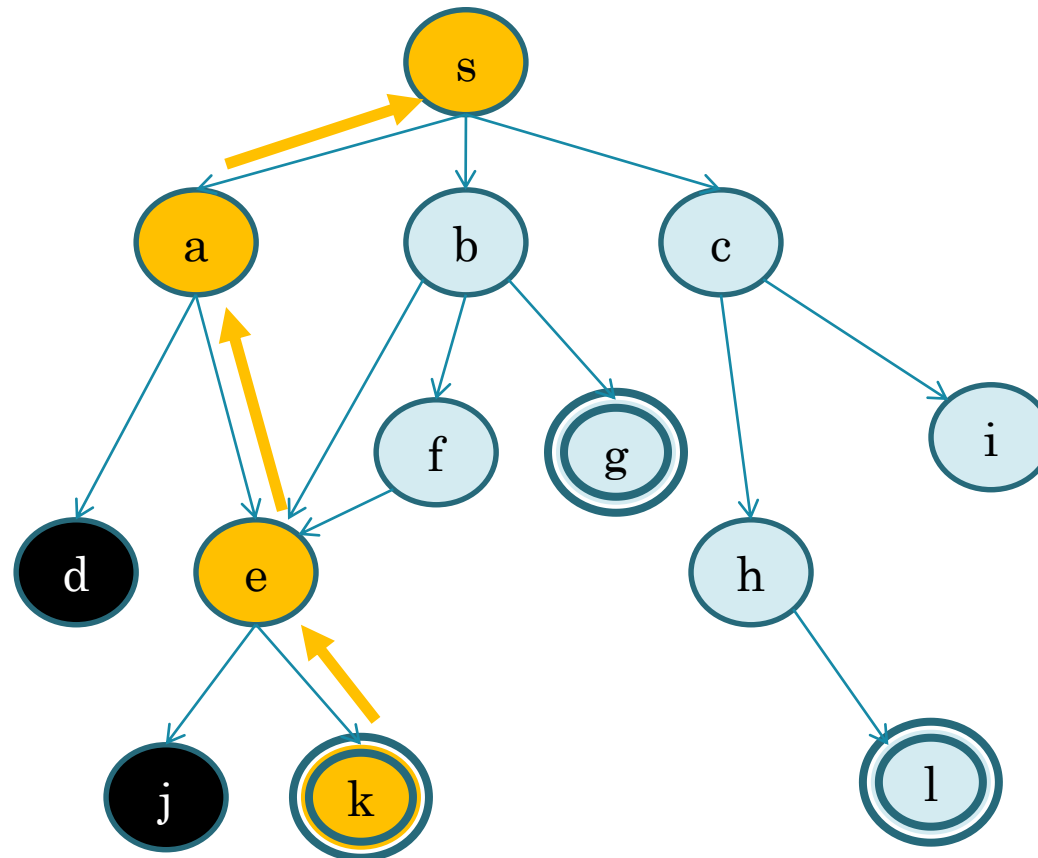
Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	-	-	a	a	-	-	-	-	e	e	-



PRIMER - ISKANJE V GLOBINO (10/10)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	F	F	T	T	F	F	F	F	T	T	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	-	-	a	a	-	-	-	-	e	e	-



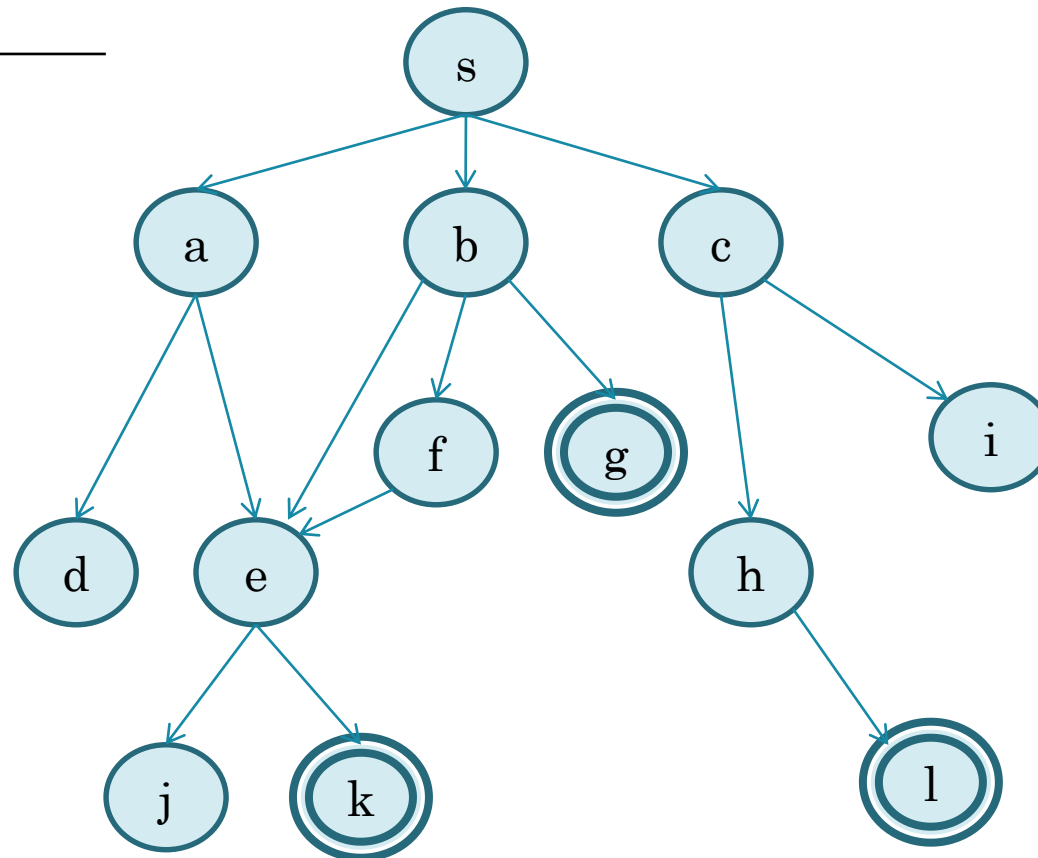
PRIMER - ISKANJE V ŠIRINO (1/11)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
F	F	F	F	F	F	F	F	F	F	F	F	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	-	-	-	-	-	-	-	-	-	-	-	-

Vrsta

.



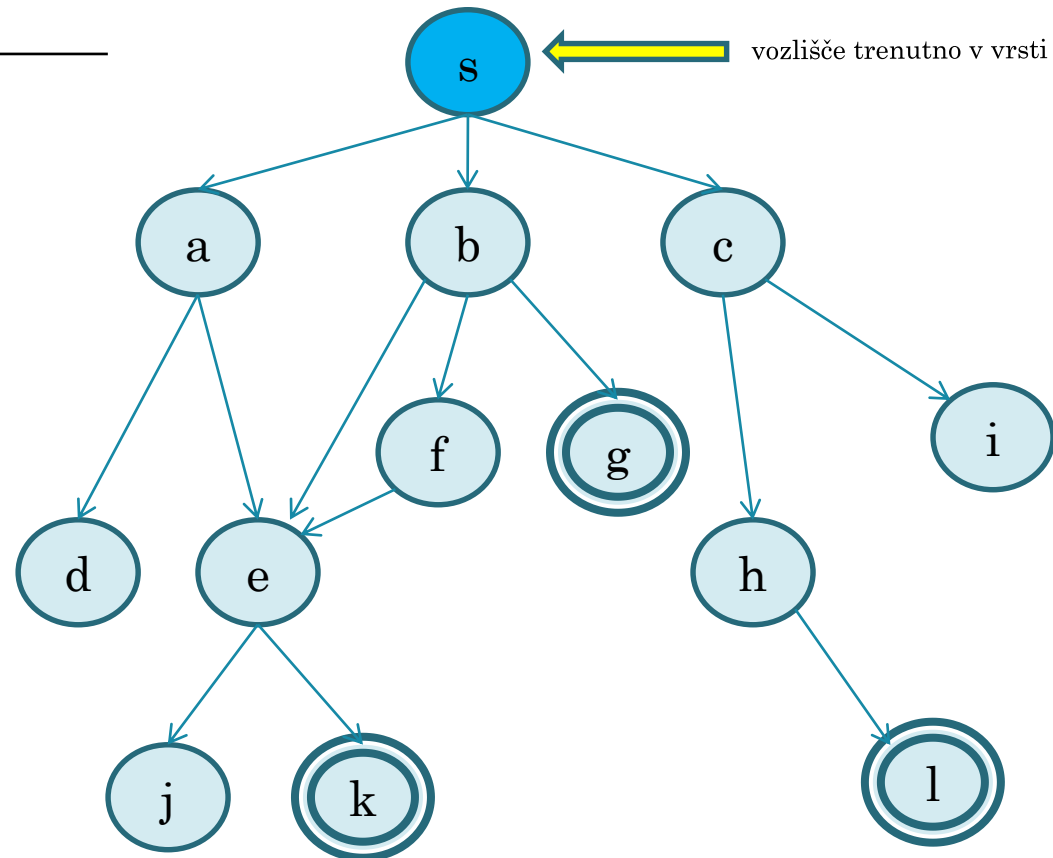
PRIMER - ISKANJE V ŠIRINO (2/11)

Obiskano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	F	F	F	F	F	F	F	F	F	F	F	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	-	-	-	-	-	-	-	-	-	-	-	-

Vrsta

s



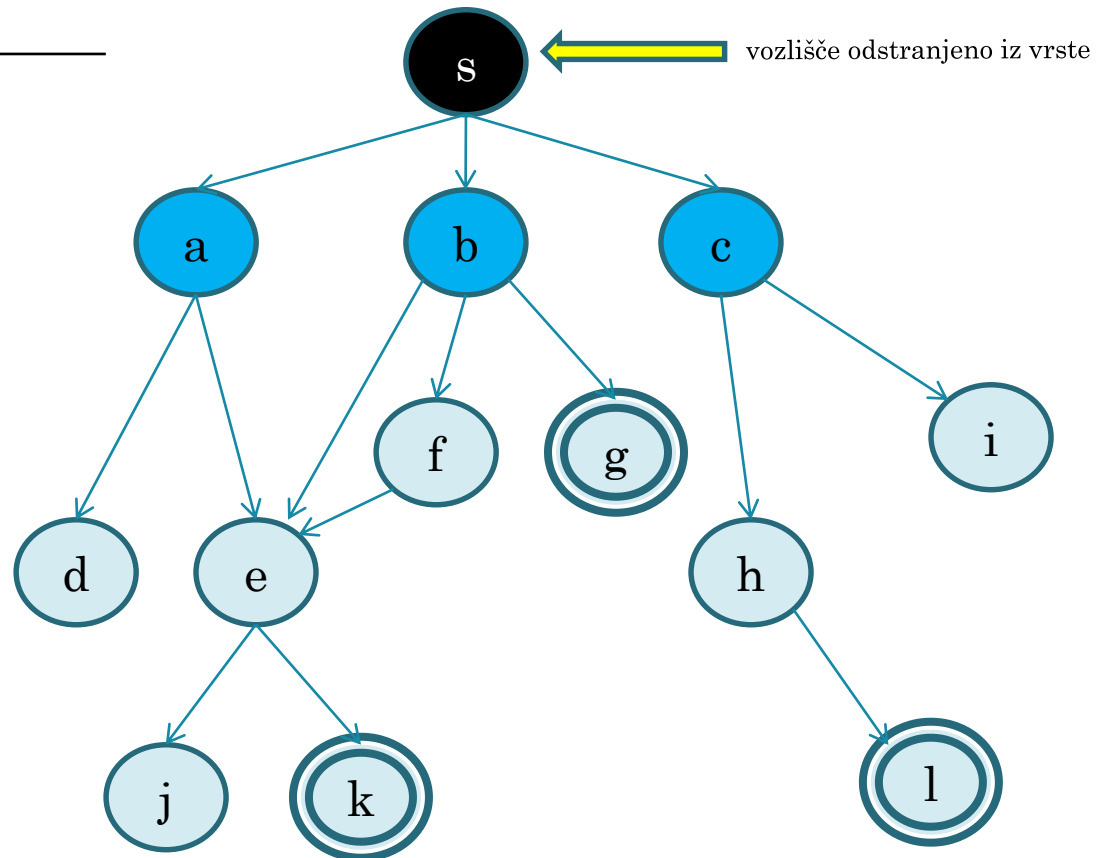
PRIMER - ISKANJE V ŠIRINO (3/11)

Dodano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	T	T	F	F	F	F	F	F	F	F	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	s	s	-	-	-	-	-	-	-	-	-

Vrsta

a	b	c
---	---	---



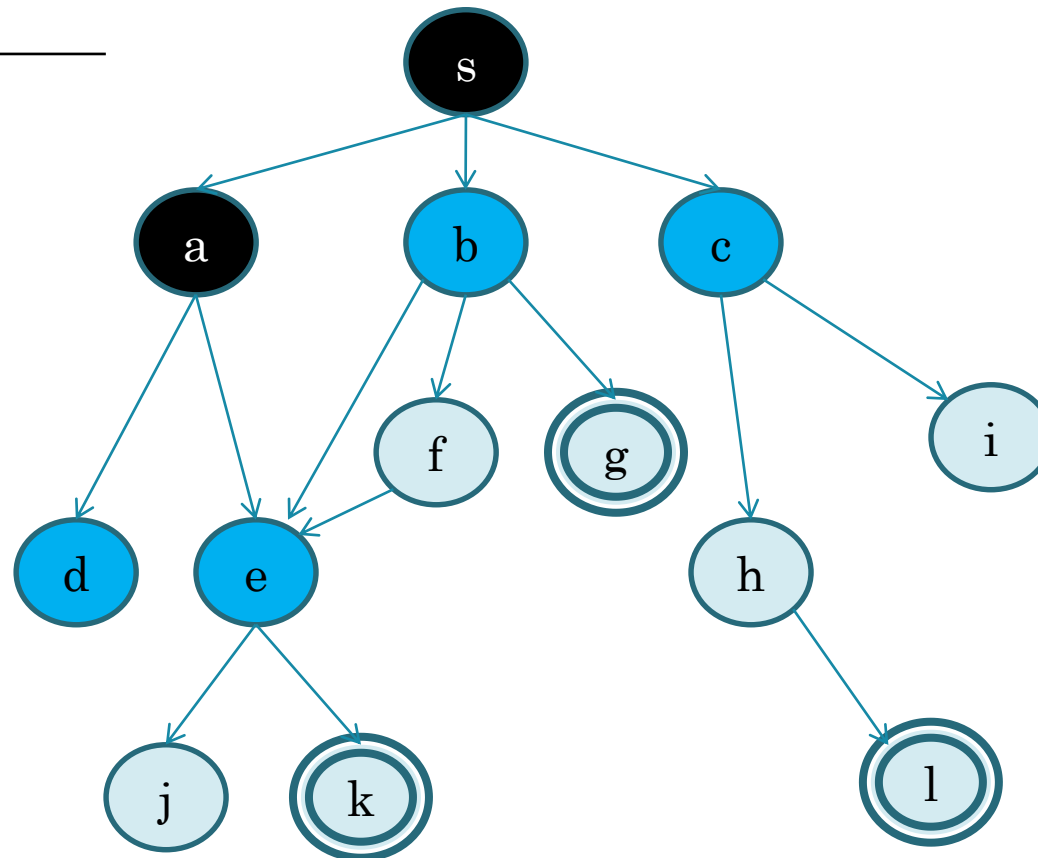
PRIMER - ISKANJE V ŠIRINO (4/11)

Dodano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	T	T	T	T	F	F	F	F	F	F	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	s	s	a	a	-	-	-	-	-	-	-

Vrsta

b	c	d	e
---	---	---	---



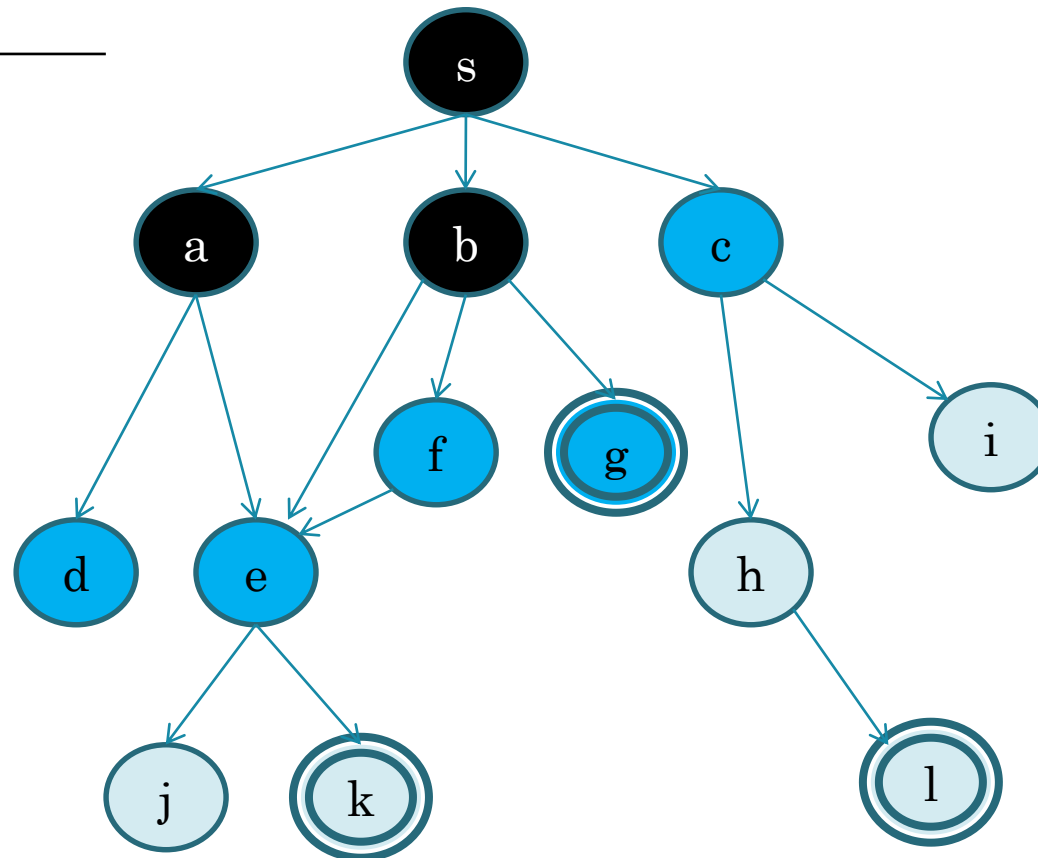
PRIMER - ISKANJE V ŠIRINO (5/11)

Dodano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	T	T	T	T	T	T	F	F	F	F	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	s	s	a	a	b	b	-	-	-	-	-

Vrsta

c	d	e	f	g
---	---	---	---	---



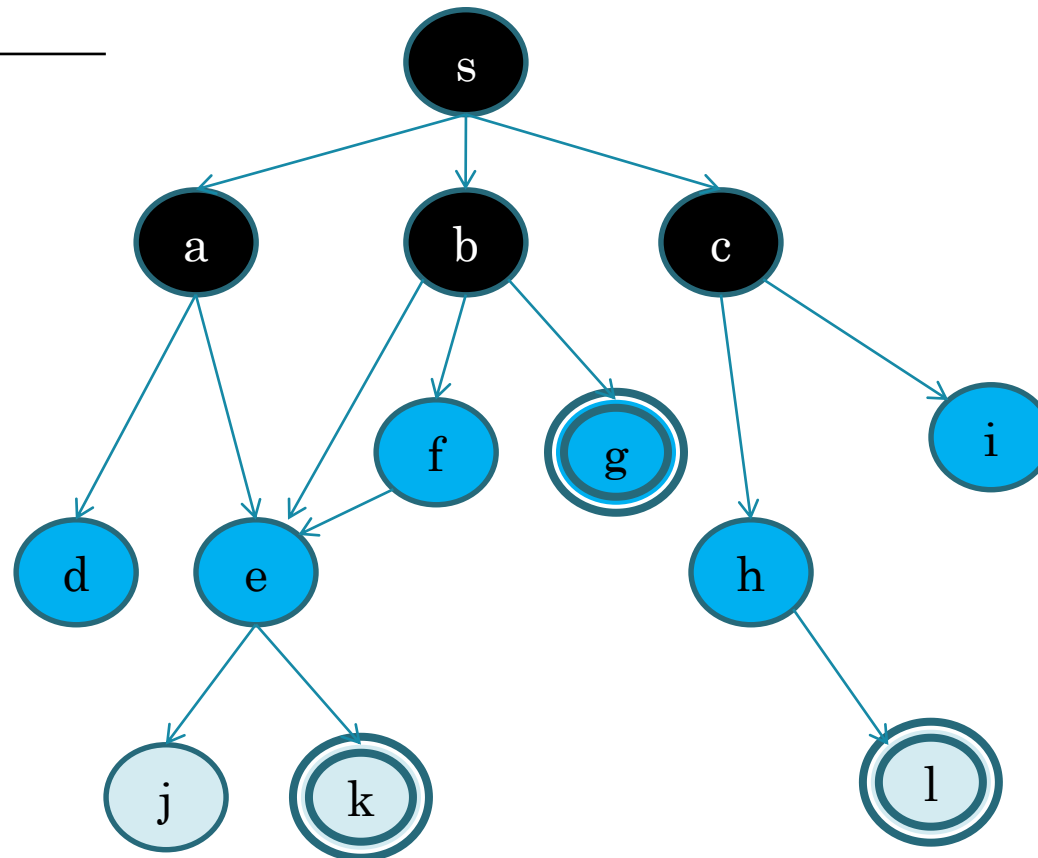
PRIMER - ISKANJE V ŠIRINO (6/11)

Dodano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	T	T	T	T	T	T	T	T	F	F	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	s	s	a	a	b	b	c	c	-	-	-

Vrsta

d	e	f	g	h	i
---	---	---	---	---	---



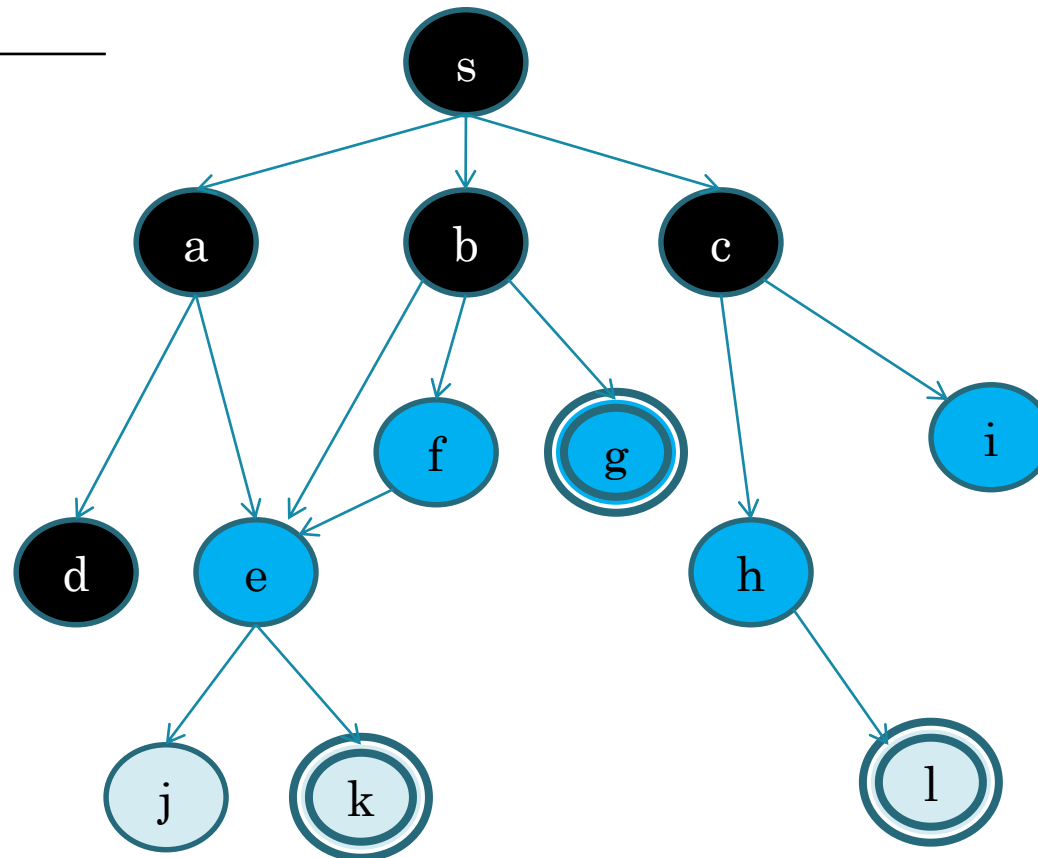
PRIMER - ISKANJE V ŠIRINO (7/11)

Dodano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	T	T	T	T	T	T	T	T	F	F	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	s	s	a	a	b	b	c	c	-	-	-

Vrsta

e	f	g	h	i
---	---	---	---	---



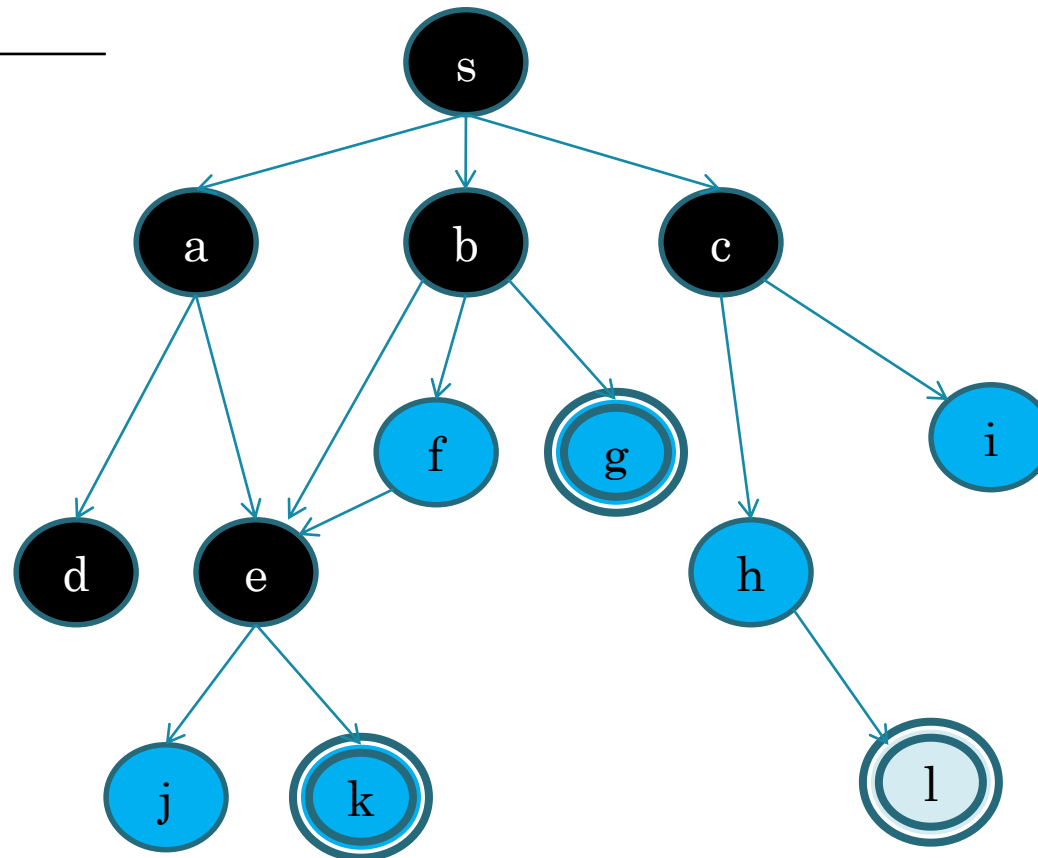
PRIMER - ISKANJE V ŠIRINO (8/11)

Dodano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	T	T	T	T	T	T	T	T	T	T	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	s	s	a	a	b	b	c	c	e	e	-

Vrsta

f	g	h	i	j	k
---	---	---	---	---	---



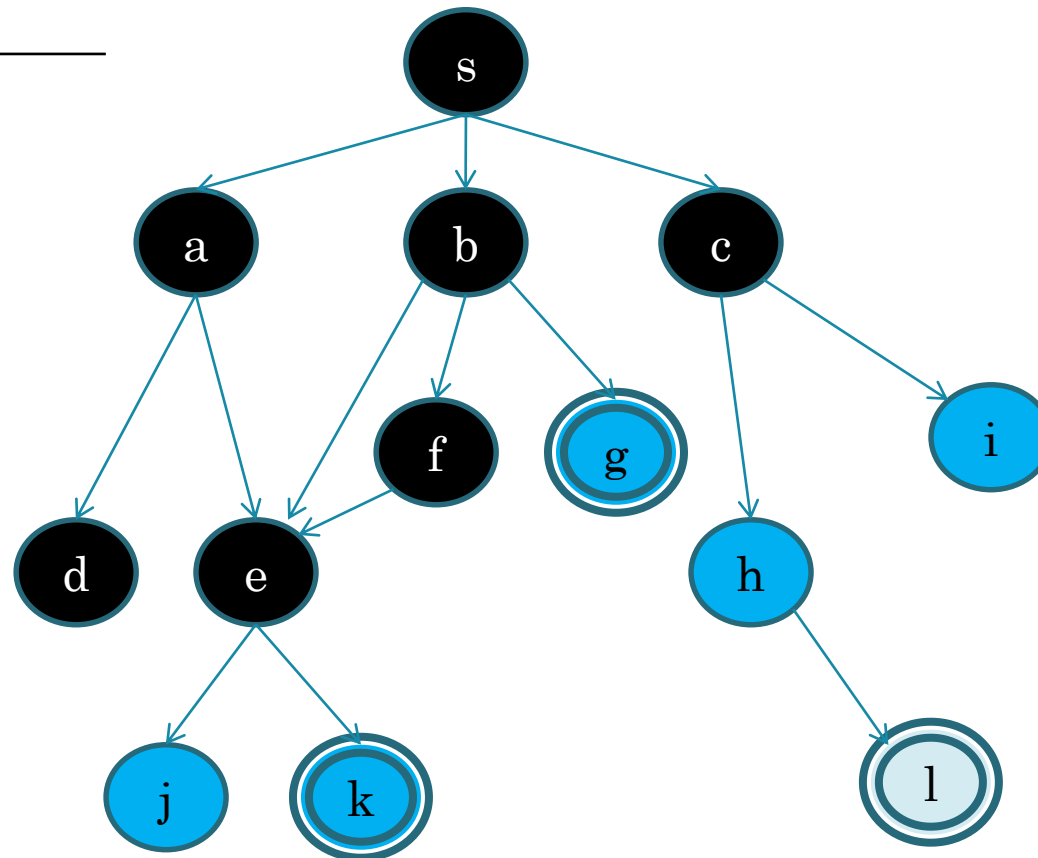
PRIMER - ISKANJE V ŠIRINO (9/11)

Dodano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	T	T	T	T	T	T	T	T	T	T	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	s	s	a	a	b	b	c	c	e	e	-

Vrsta

g	h	i	j	k
---	---	---	---	---



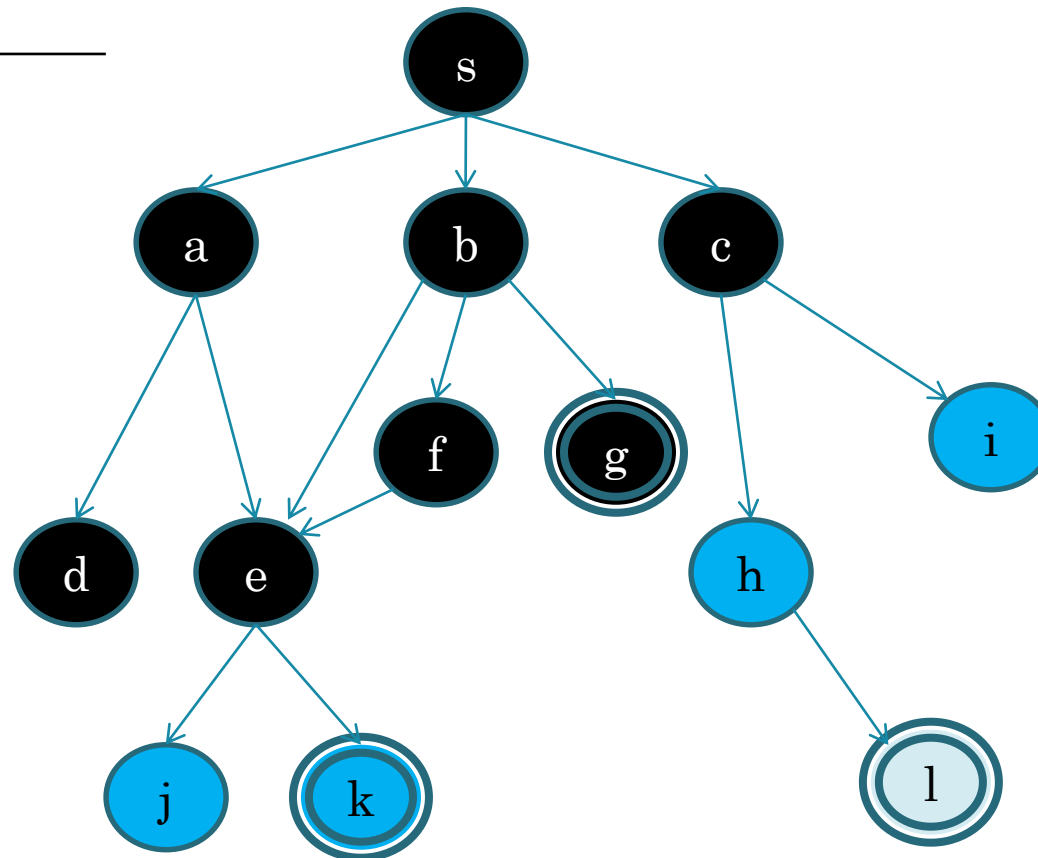
PRIMER - ISKANJE V ŠIRINO (10/11)

Dodano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	T	T	T	T	T	T	T	T	T	T	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	s	s	a	a	b	b	c	c	e	e	-

Vrsta

h	i	j	k
---	---	---	---



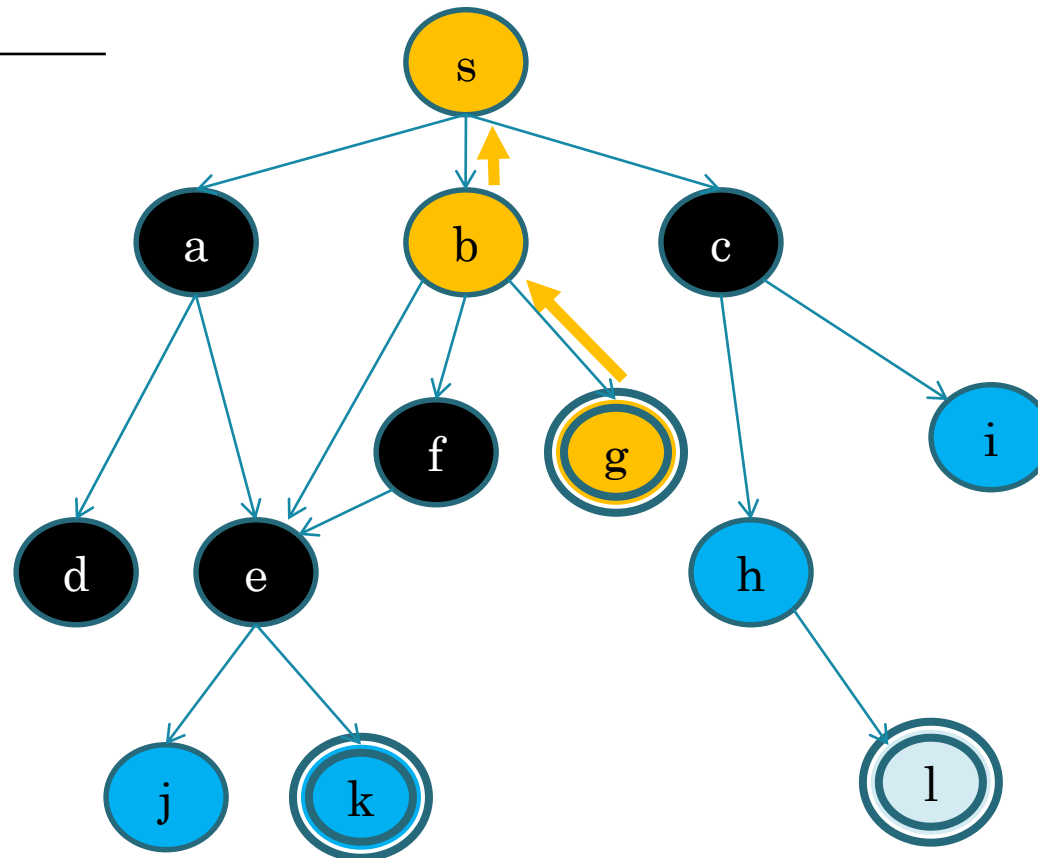
PRIMER - ISKANJE V ŠIRINO (11/11)

Dodano												
s	a	b	c	d	e	f	g	h	i	j	k	l
T	T	T	T	T	T	T	T	T	T	T	T	F

Starš na poti do rešitve												
s	a	b	c	d	e	f	g	h	i	j	k	l
-	s	s	s	a	a	b	b	c	c	e	e	-

Vrsta

h	i	j	k
---	---	---	---



NEINFORMIRANO PREISKOVANJE

Iterativno poglobljanje

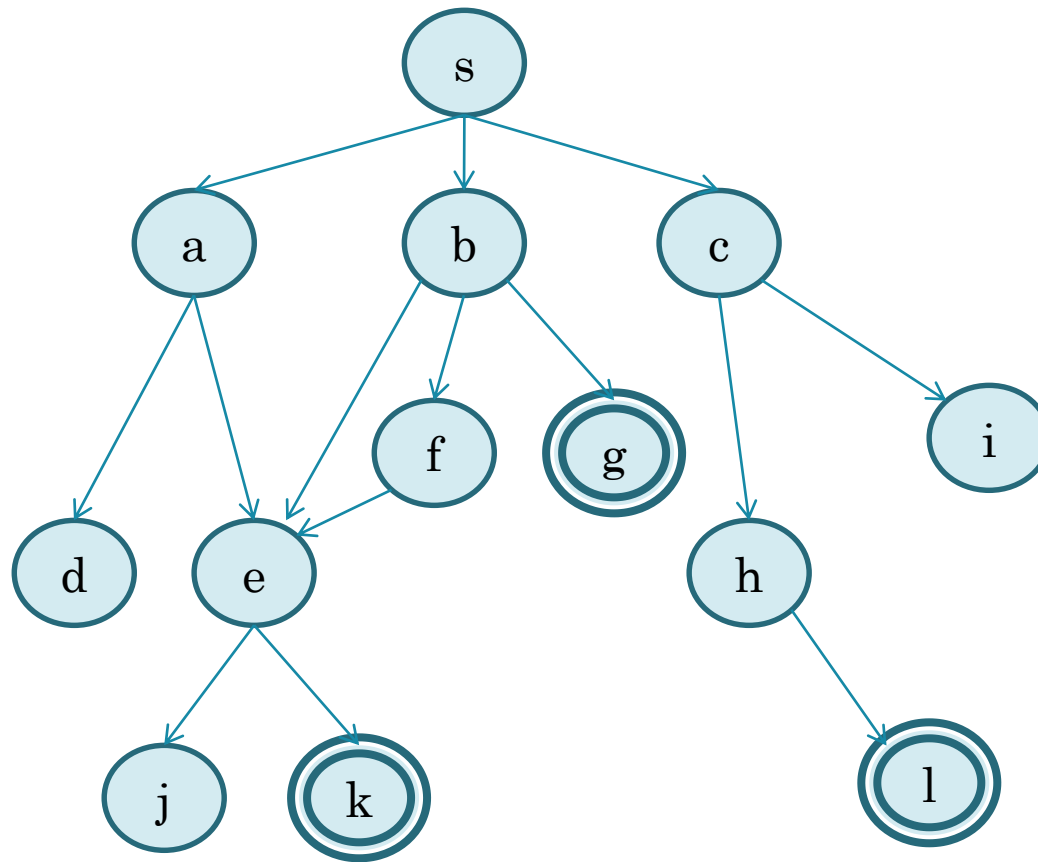
- iskanje v globino z omejeno globino, ki jo iterativno podaljšujemo
 - kombinira prednosti iskanja v globino in iskanja v širino
 - pomnilniško manj zahtevno
 - vedno najprej najde najkrajšo pot
 - časovna zahtevnost reda $O(b^d)$, prostorska zahtevnost reda $O(bd)$

b – faktor vejanja grafa

d – globina najbližjega končnega stanja

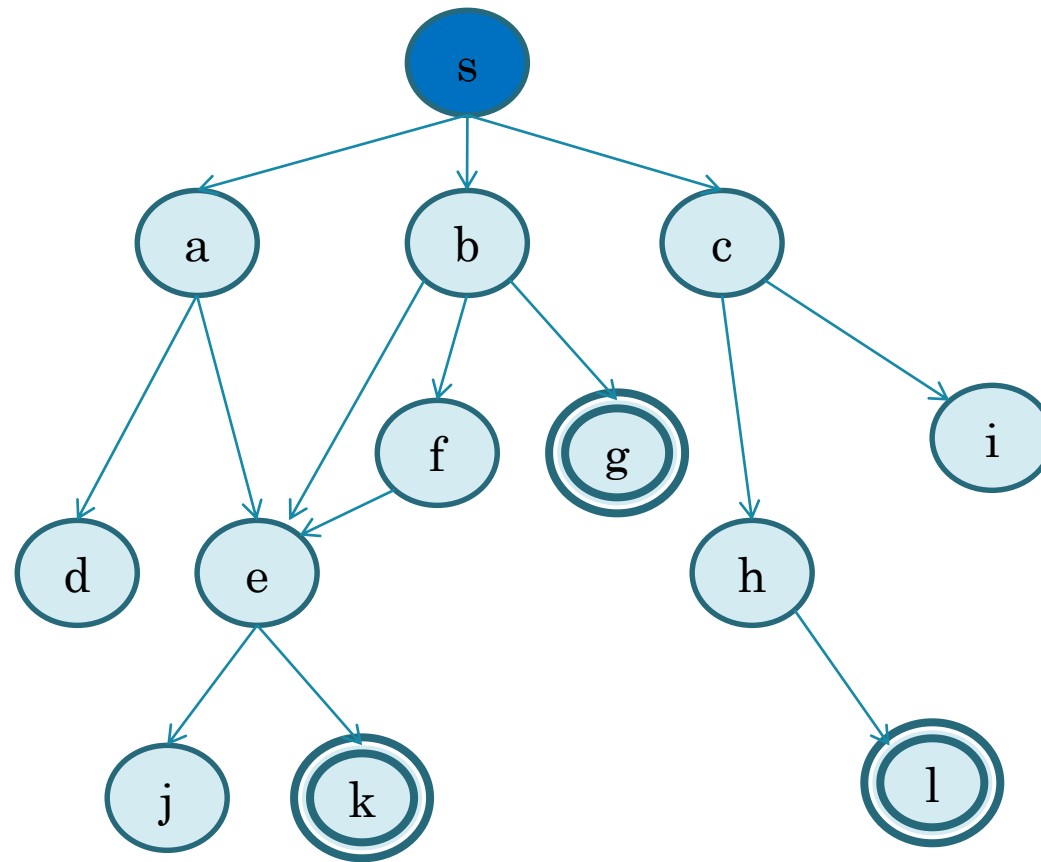
PRIMER – ITERATIVNO POGLABLJANJE (1/16)

Meja 0



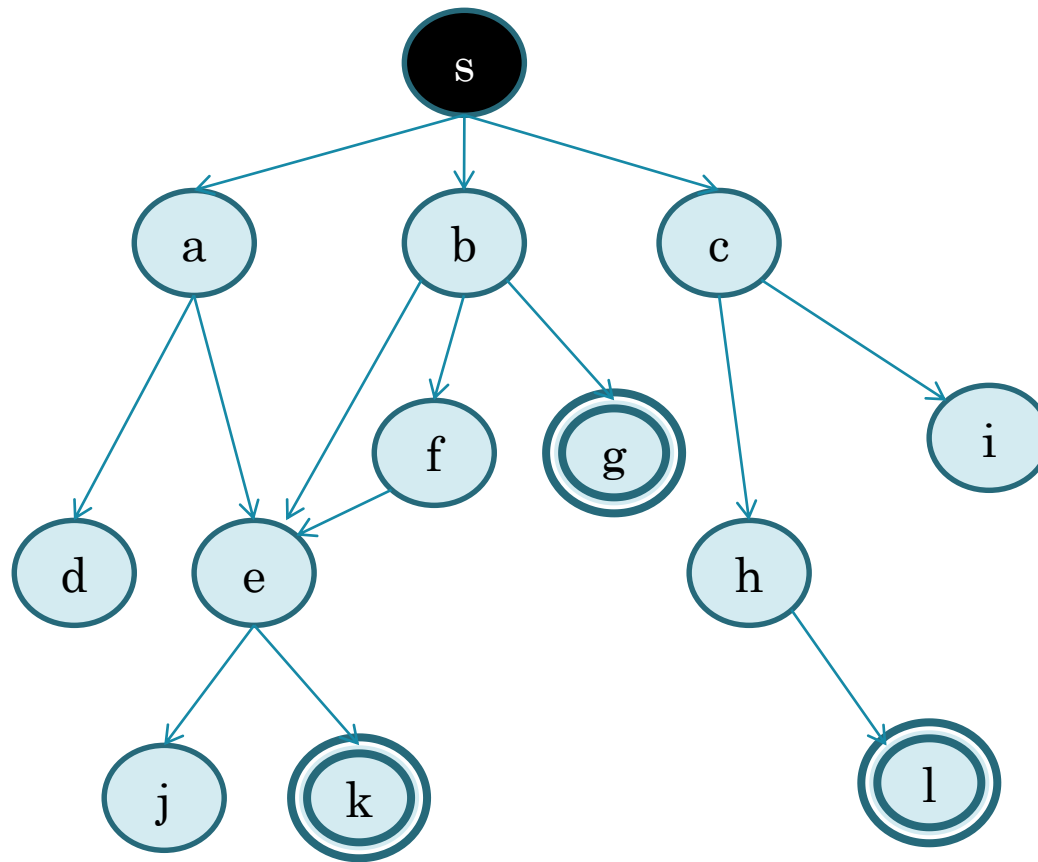
PRIMER – ITERATIVNO POGLABLJANJE (2/16)

Meja 0



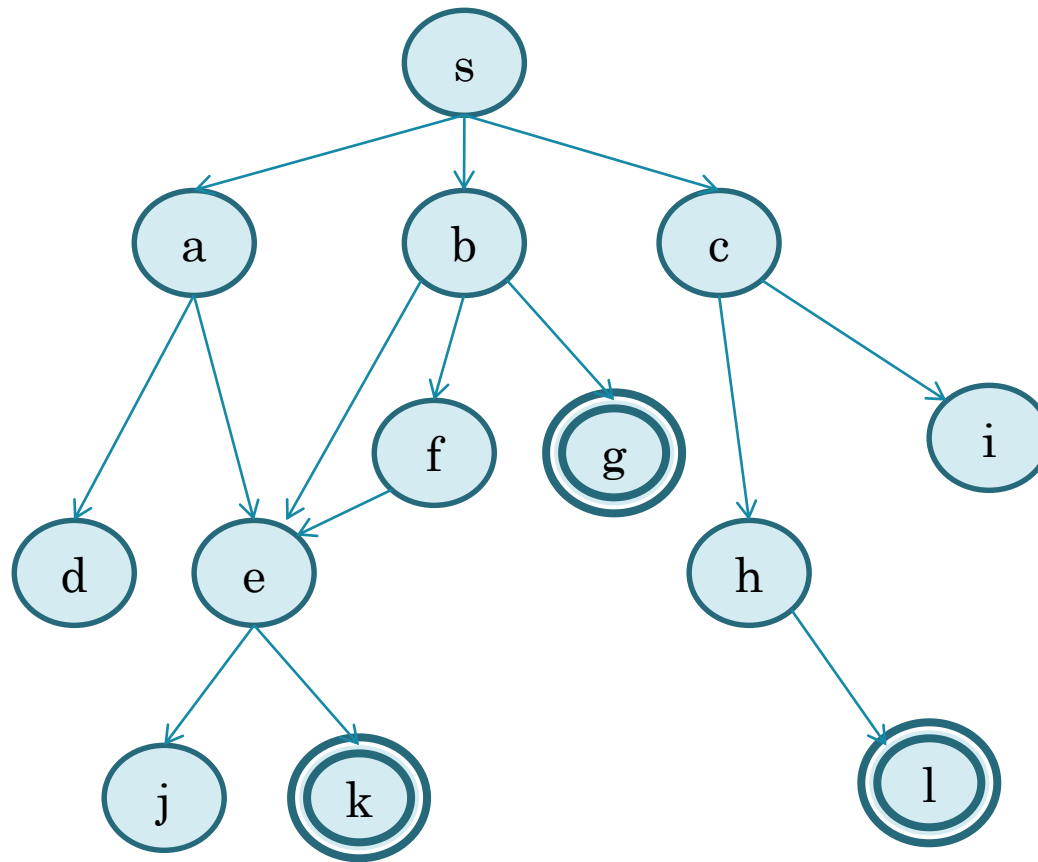
PRIMER – ITERATIVNO POGLABLJANJE (3/16)

Meja 0



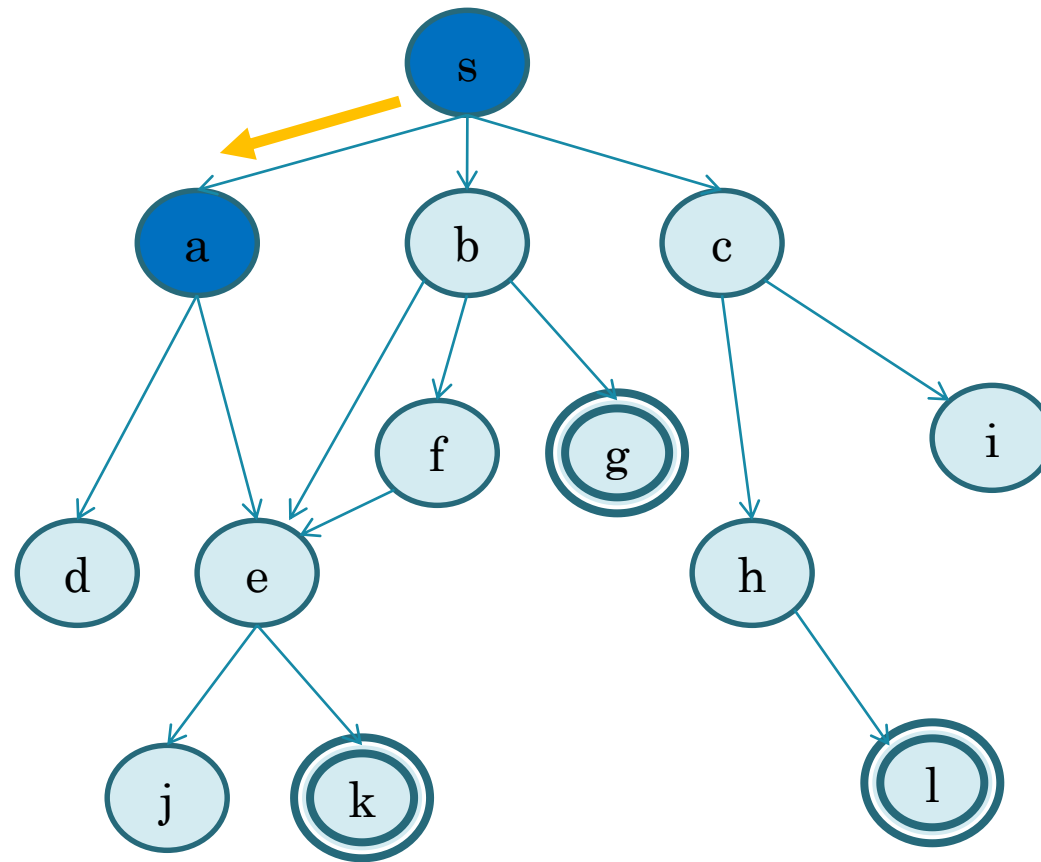
PRIMER – ITERATIVNO POGLABLJANJE (4/16)

Meja 1



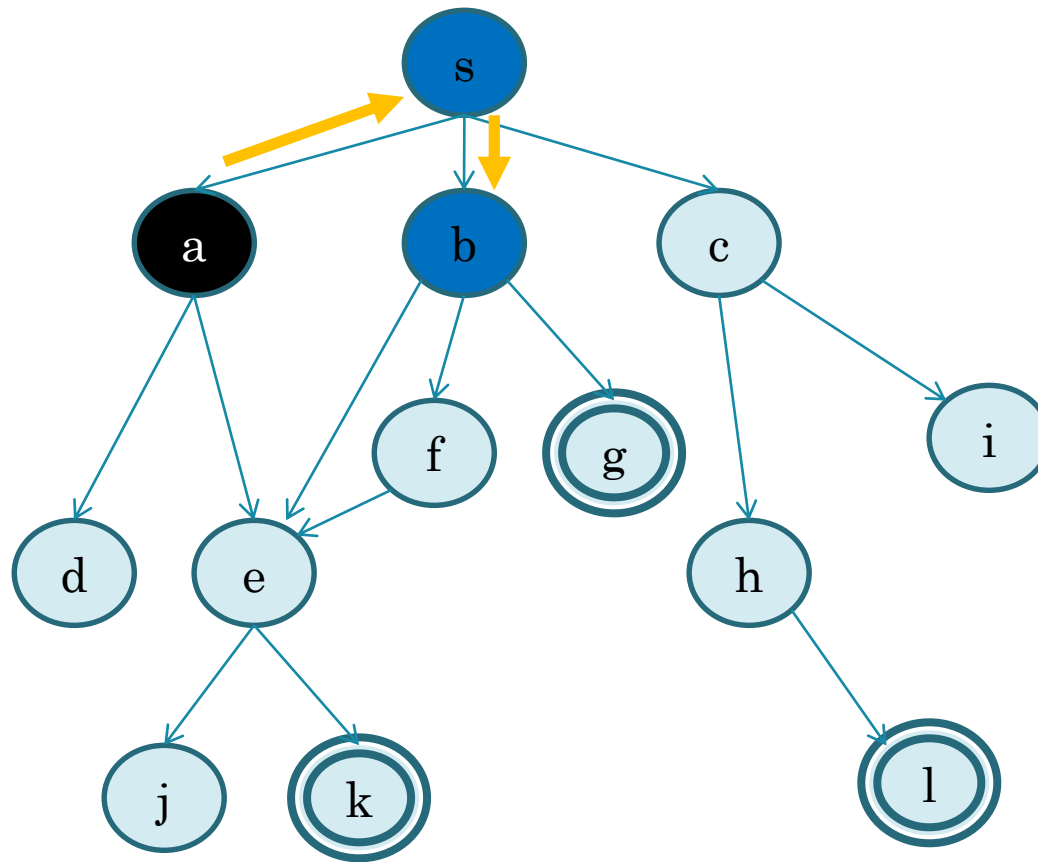
PRIMER – ITERATIVNO POGLABLJANJE (5/16)

Meja 1



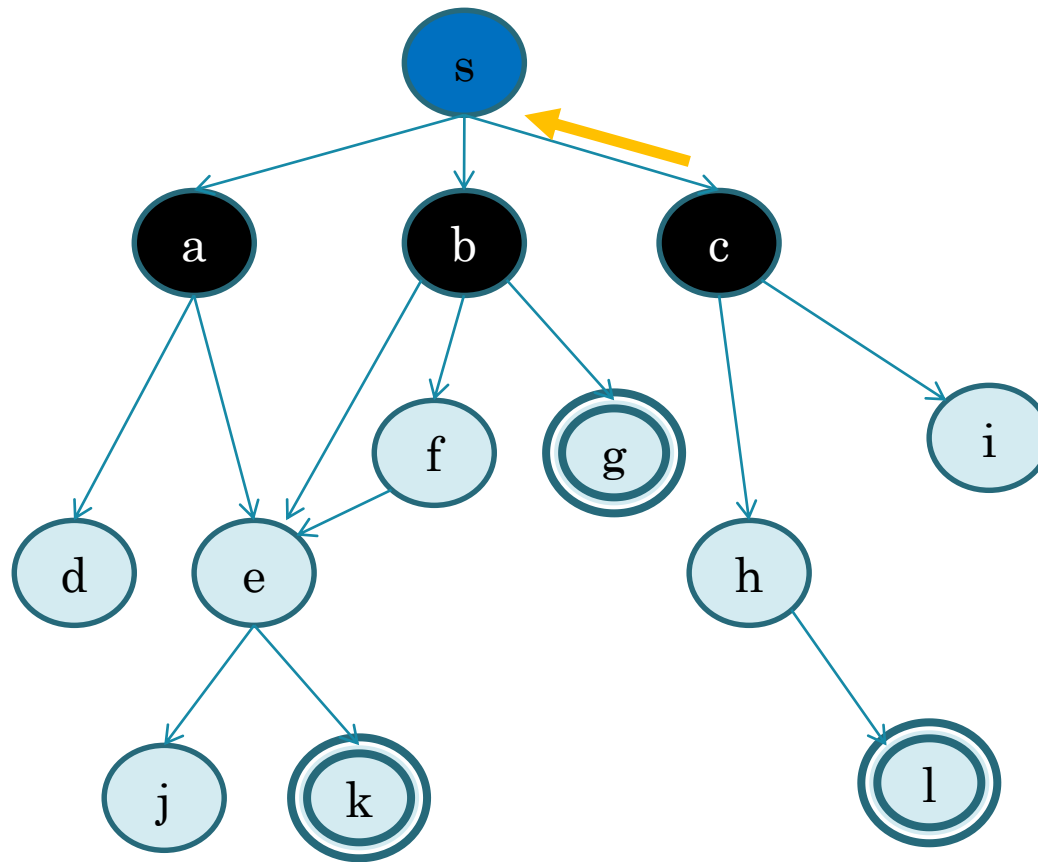
PRIMER – ITERATIVNO POGLABLJANJE (6/16)

Meja 1



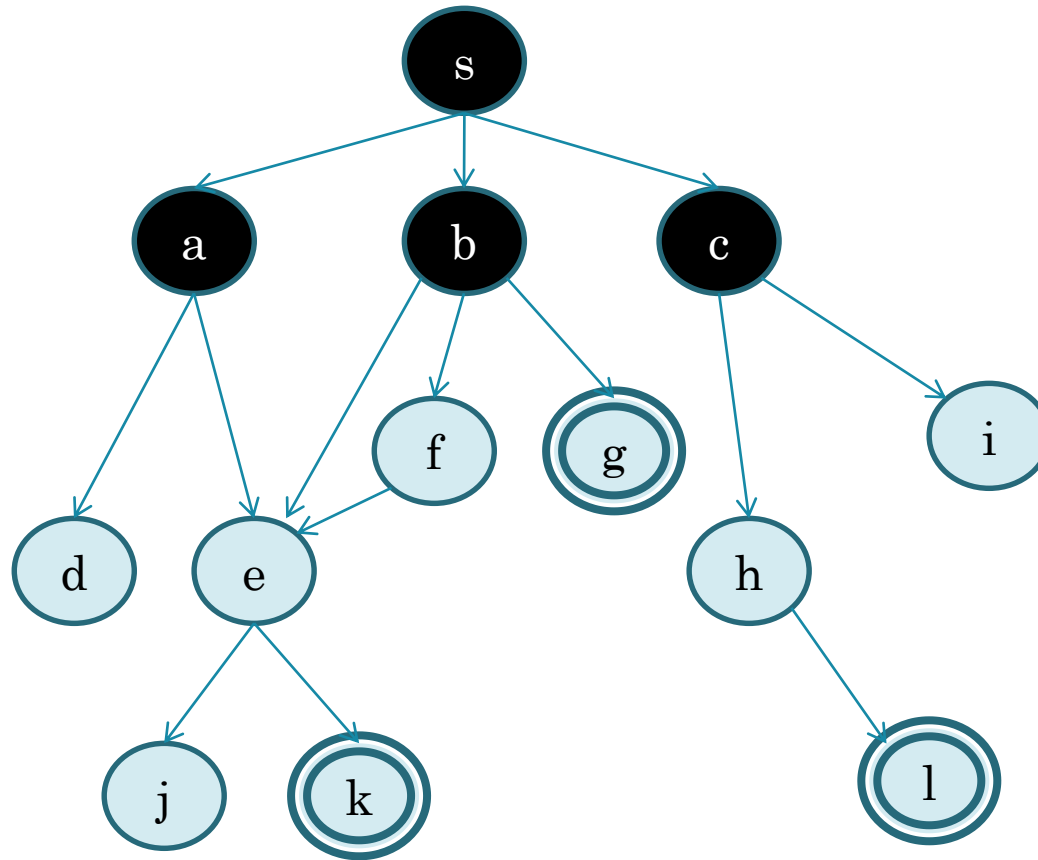
PRIMER – ITERATIVNO POGLABLJANJE (8/16)

Meja 1



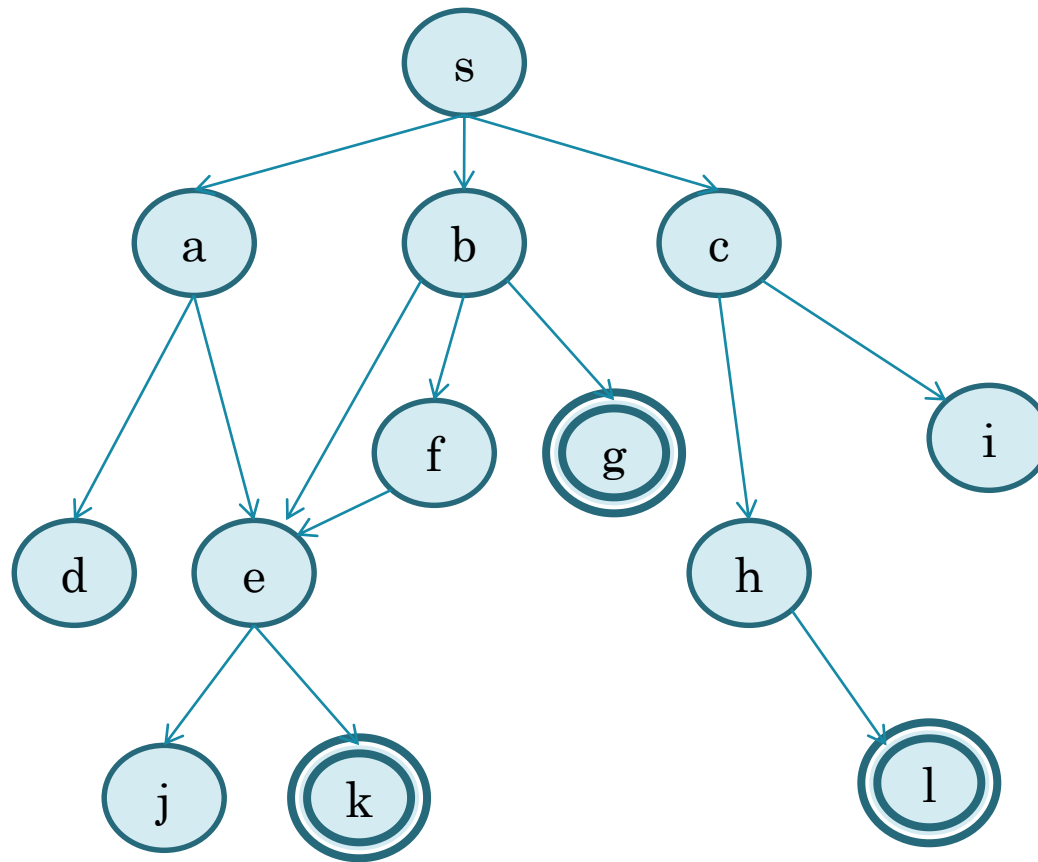
PRIMER – ITERATIVNO POGLABLJANJE (9/16)

Meja 1



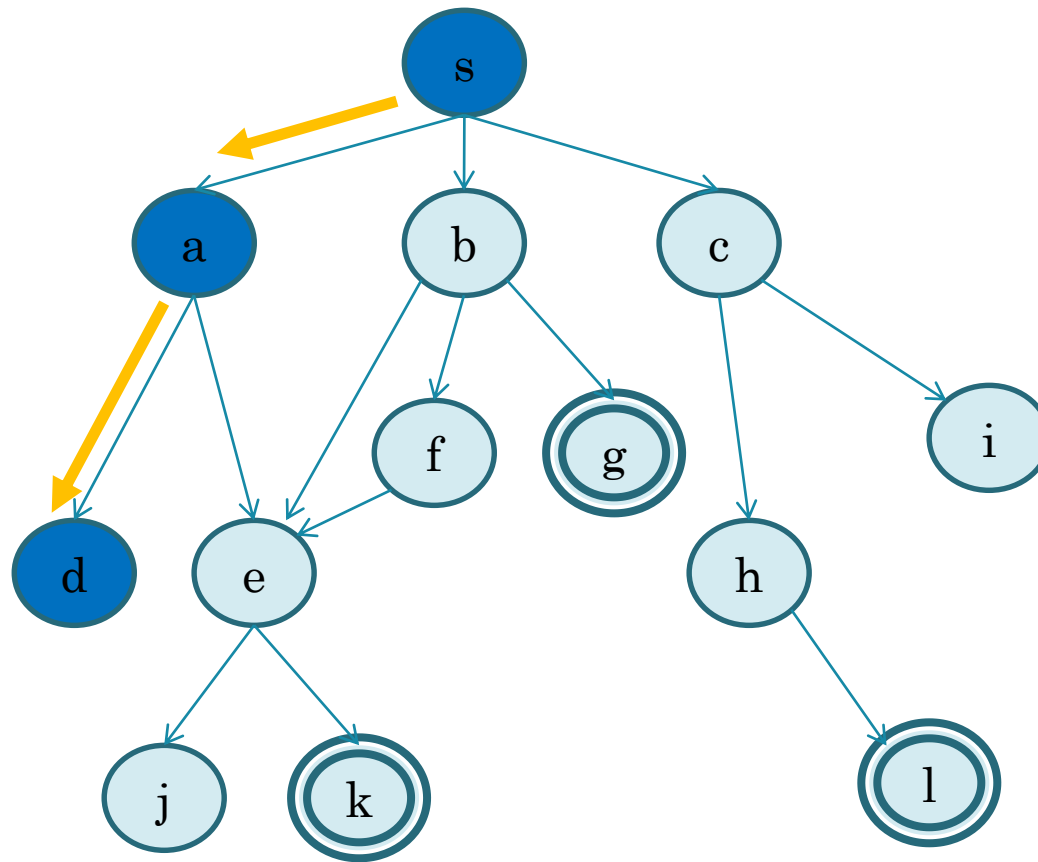
PRIMER – ITERATIVNO POGLABLJANJE (10/16)

Meja 2



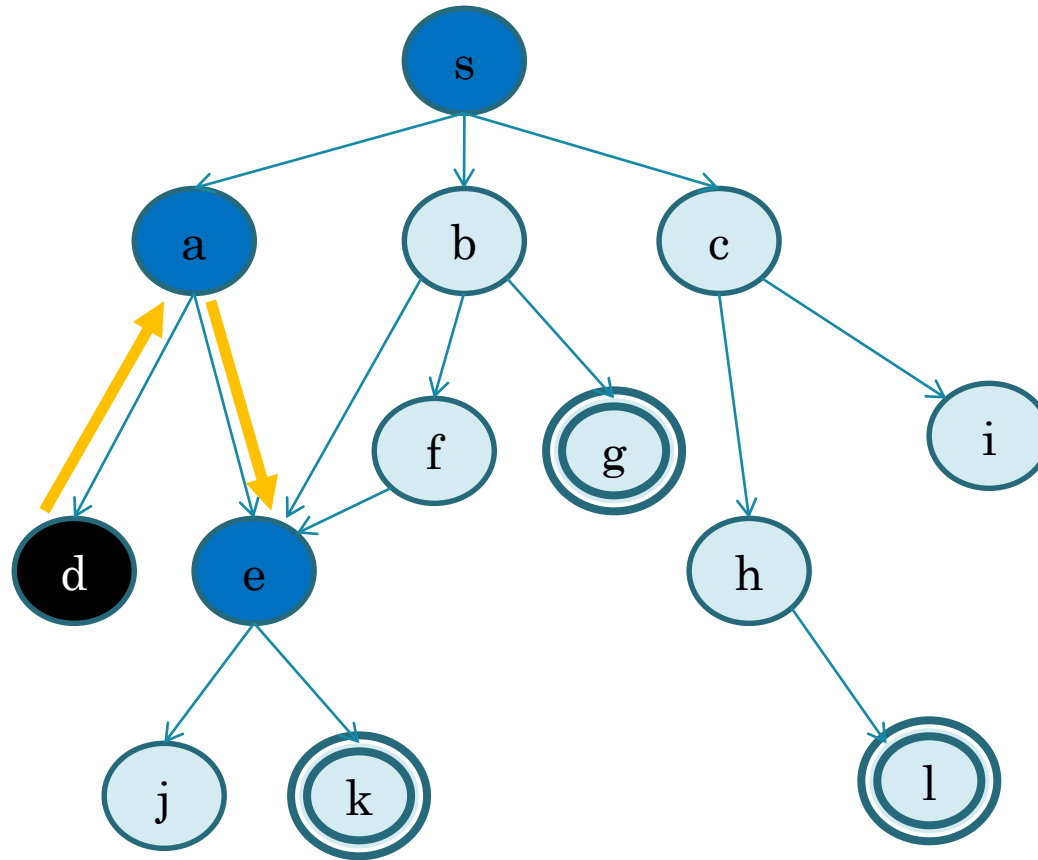
PRIMER – ITERATIVNO POGLABLJANJE (11/16)

Meja 2



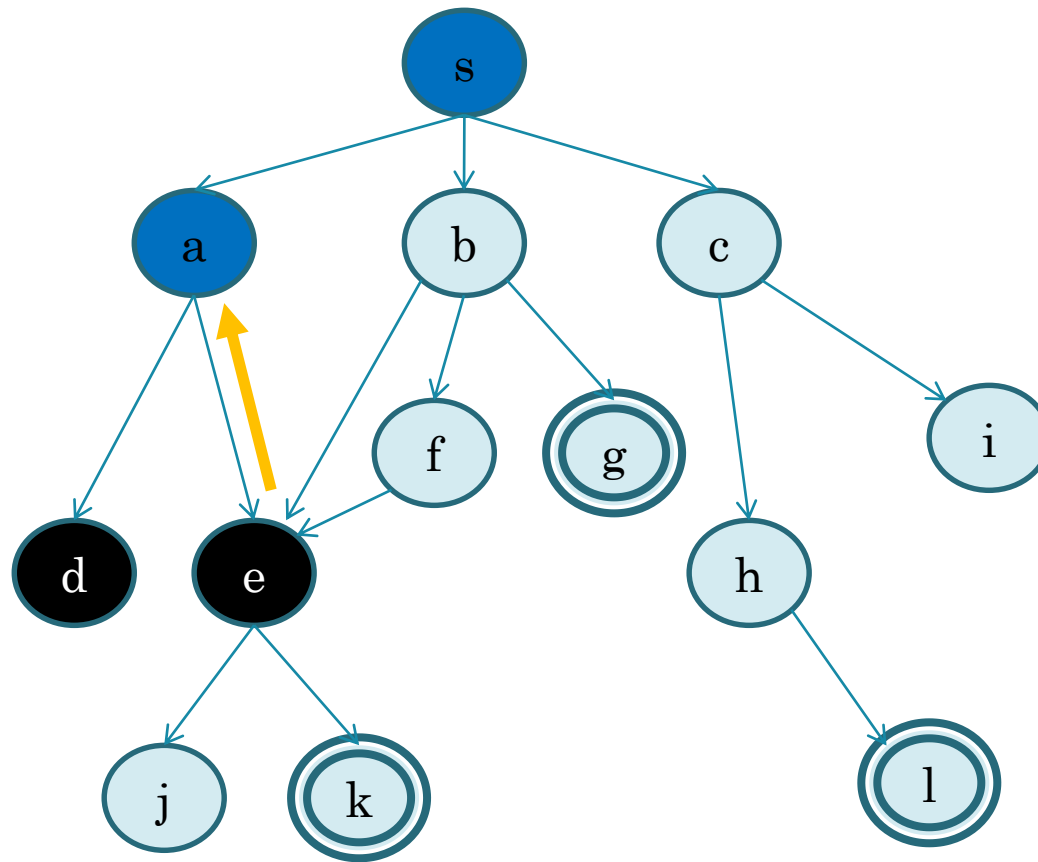
PRIMER – ITERATIVNO POGLABLJANJE (12/16)

Meja 2



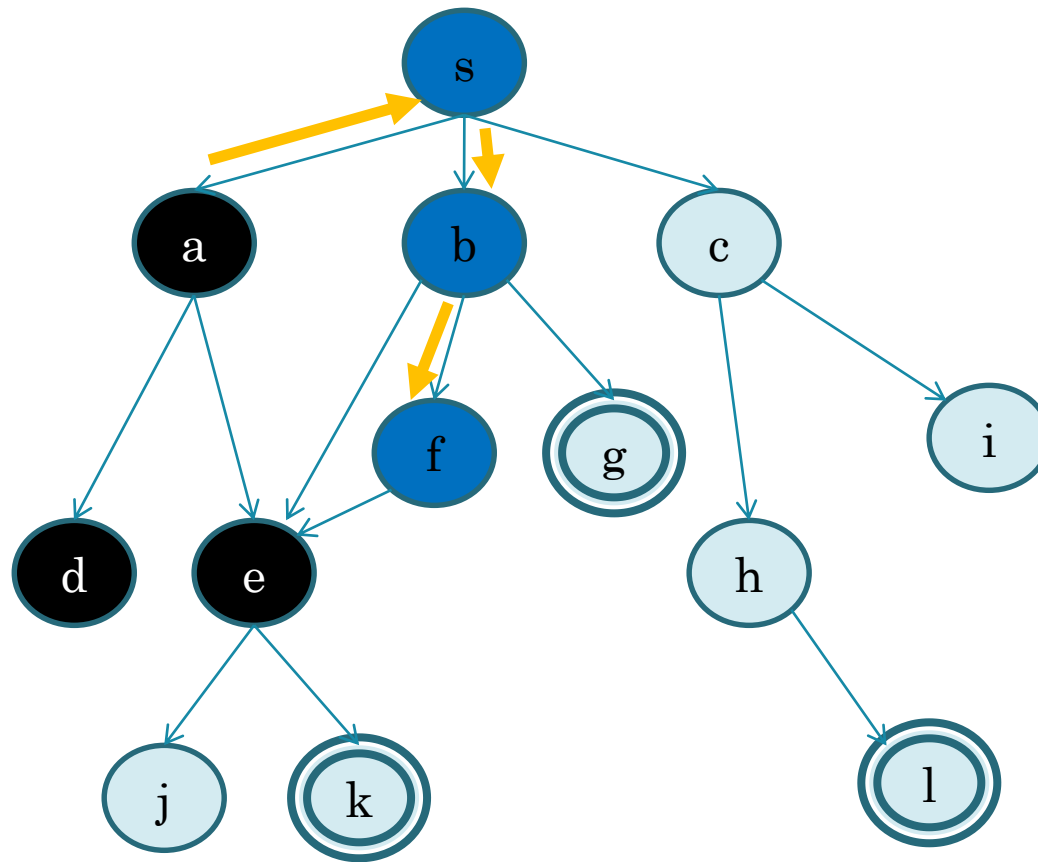
PRIMER – ITERATIVNO POGLABLJANJE (13/16)

Meja 2



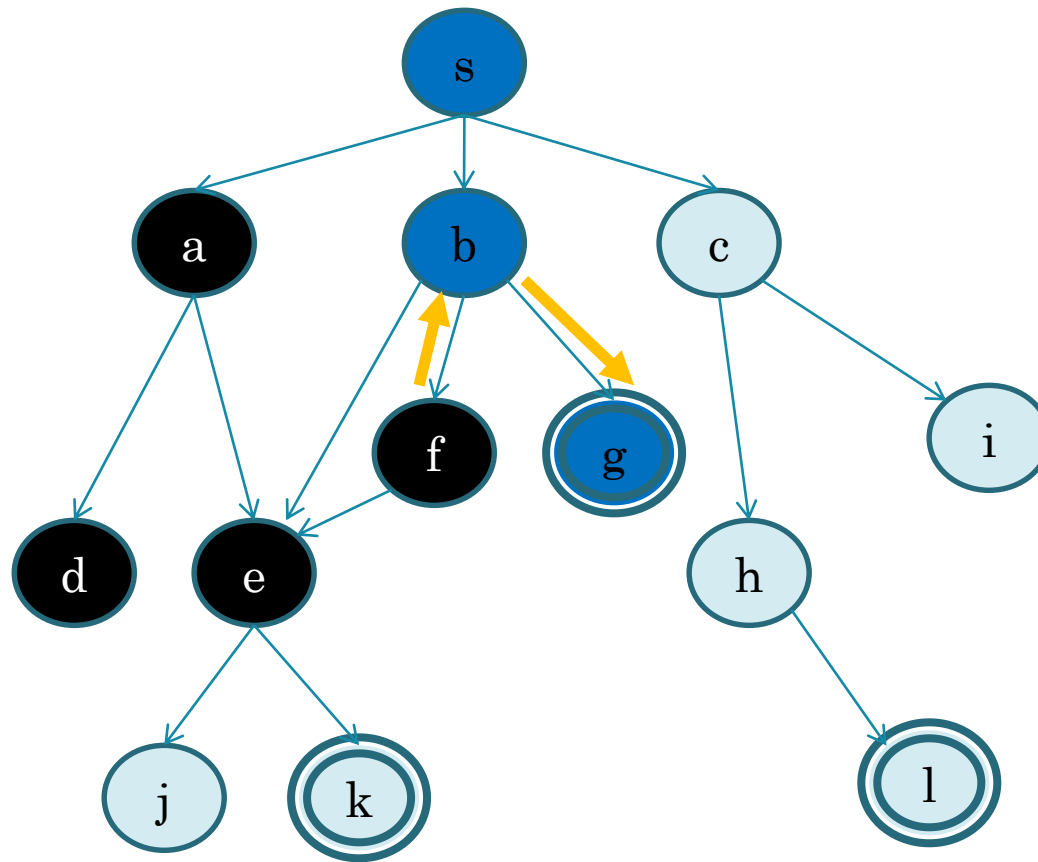
PRIMER – ITERATIVNO POGLABLJANJE (14/16)

Meja 2



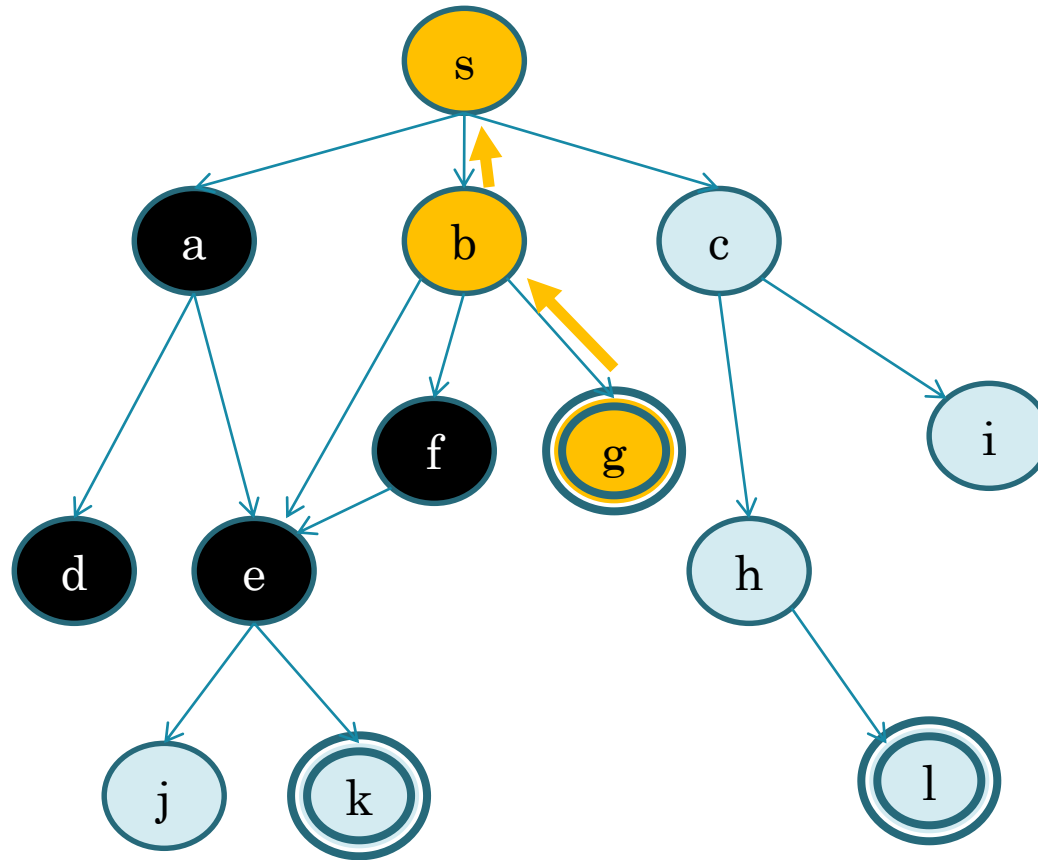
PRIMER – ITERATIVNO POGLABLJANJE (15/16)

Meja 2



PRIMER – ITERATIVNO POGLABLJANJE (16/16)

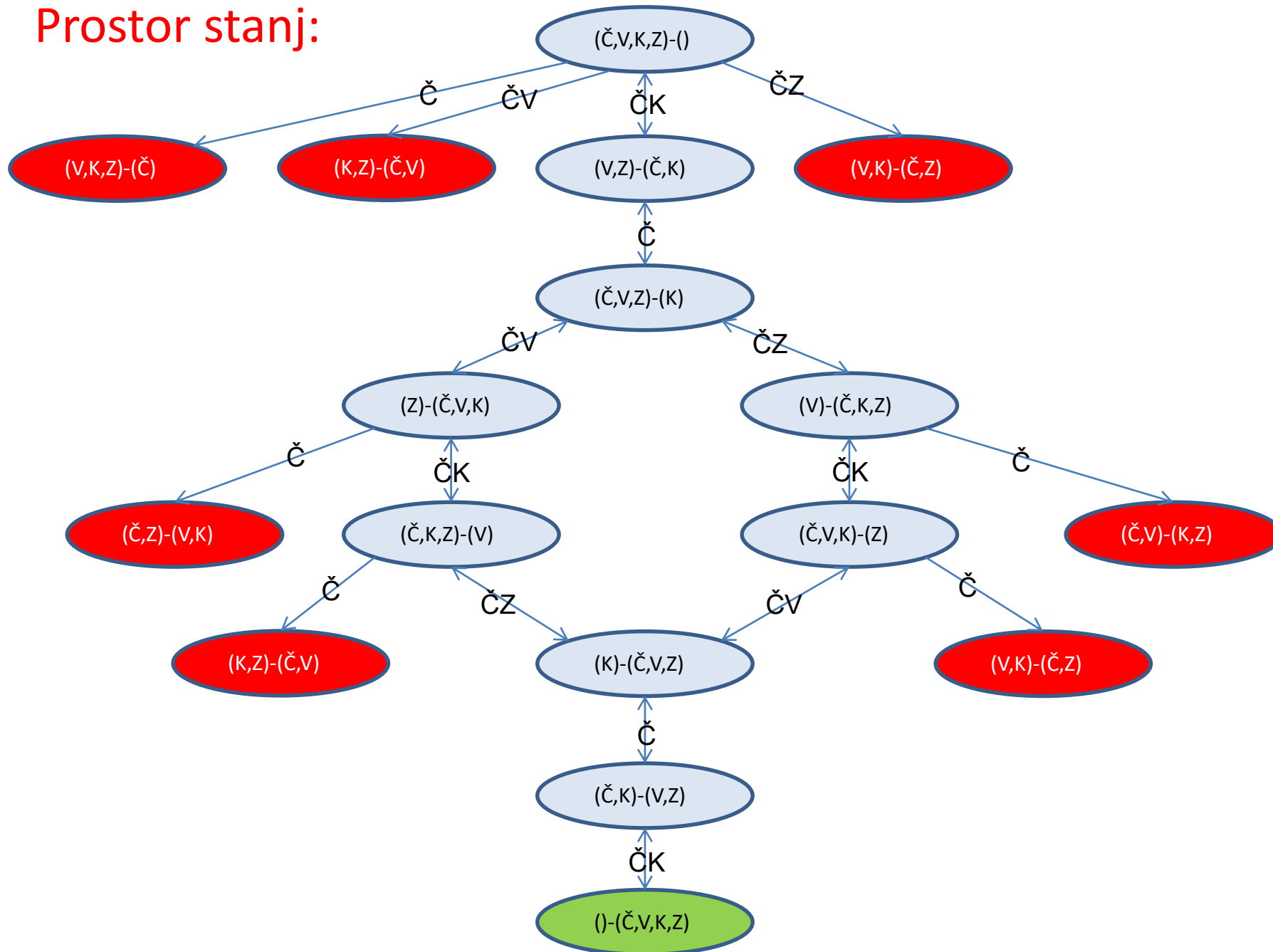
Meja 2



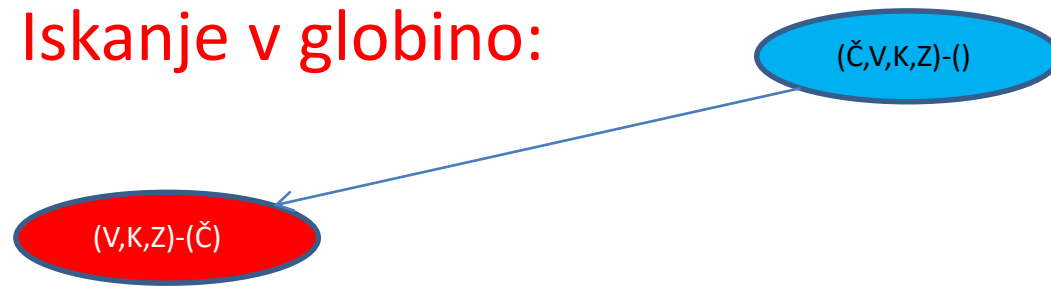
PRIMER – VOLK, KOZA, ZELJE



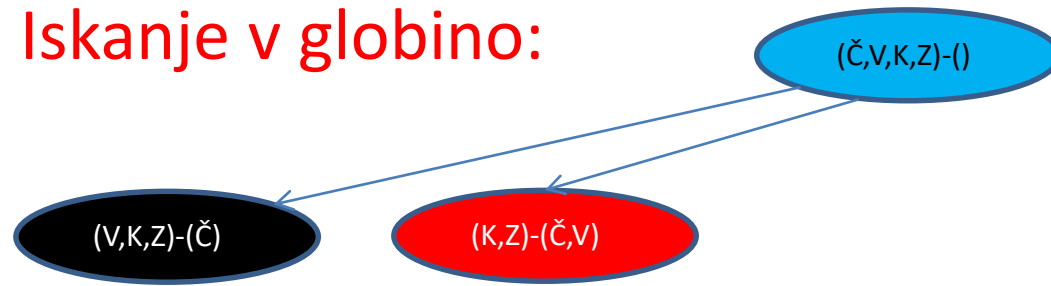
Prostor stanj:



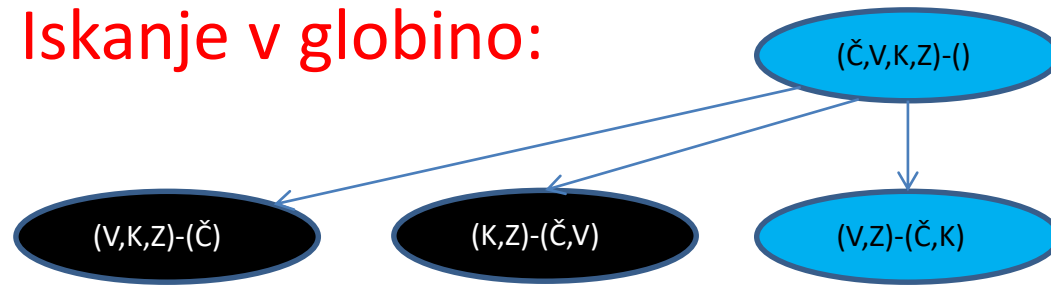
Iskanje v globino:



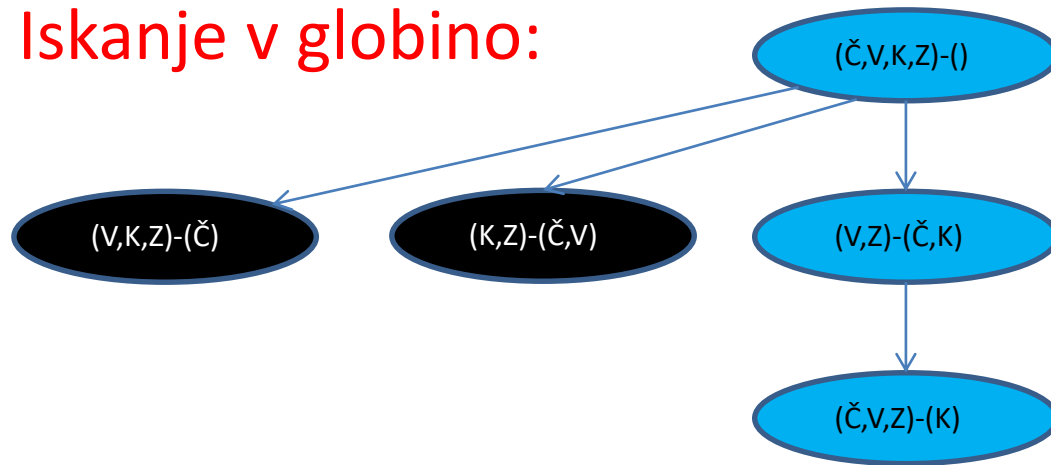
Iskanje v globino:



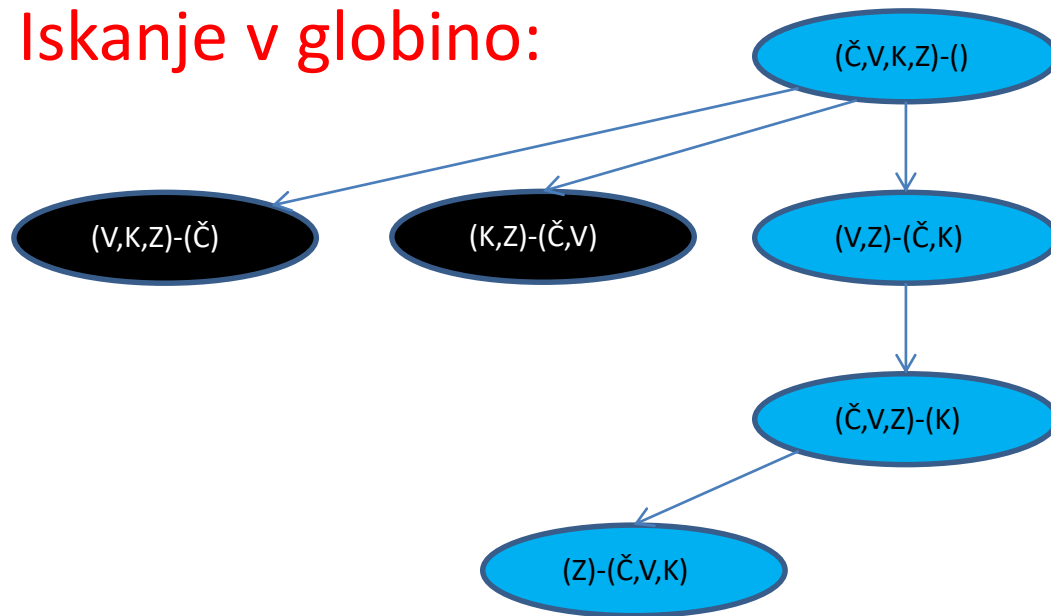
Iskanje v globino:



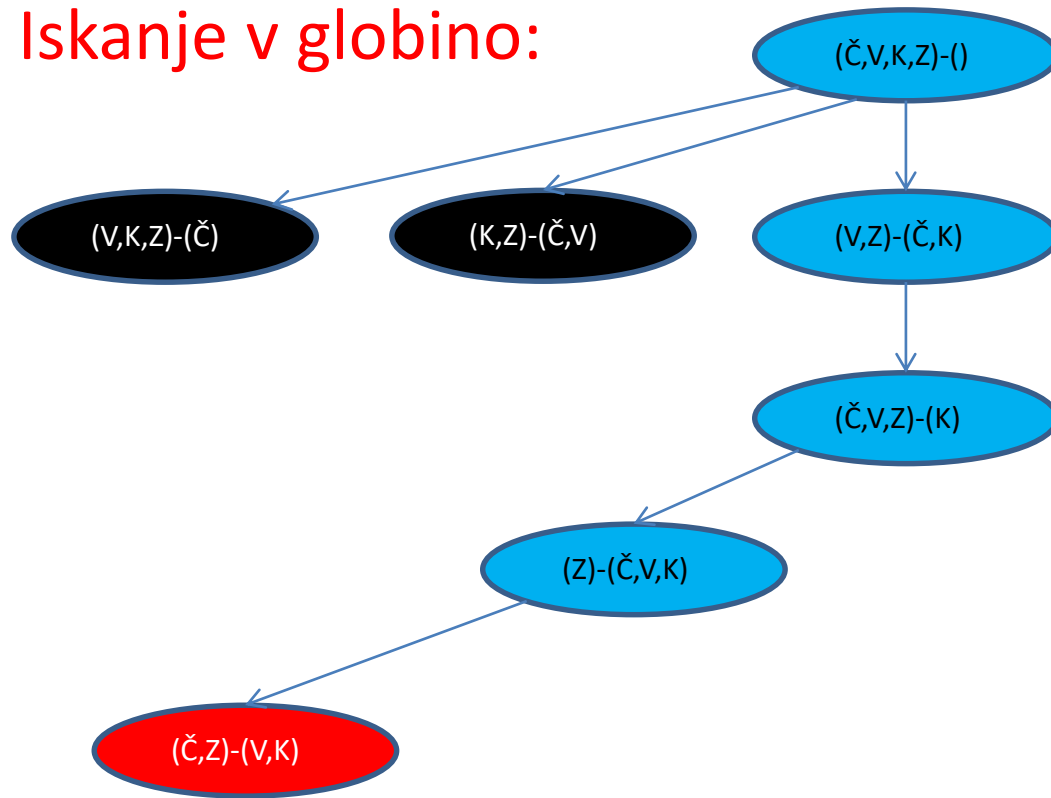
Iskanje v globino:



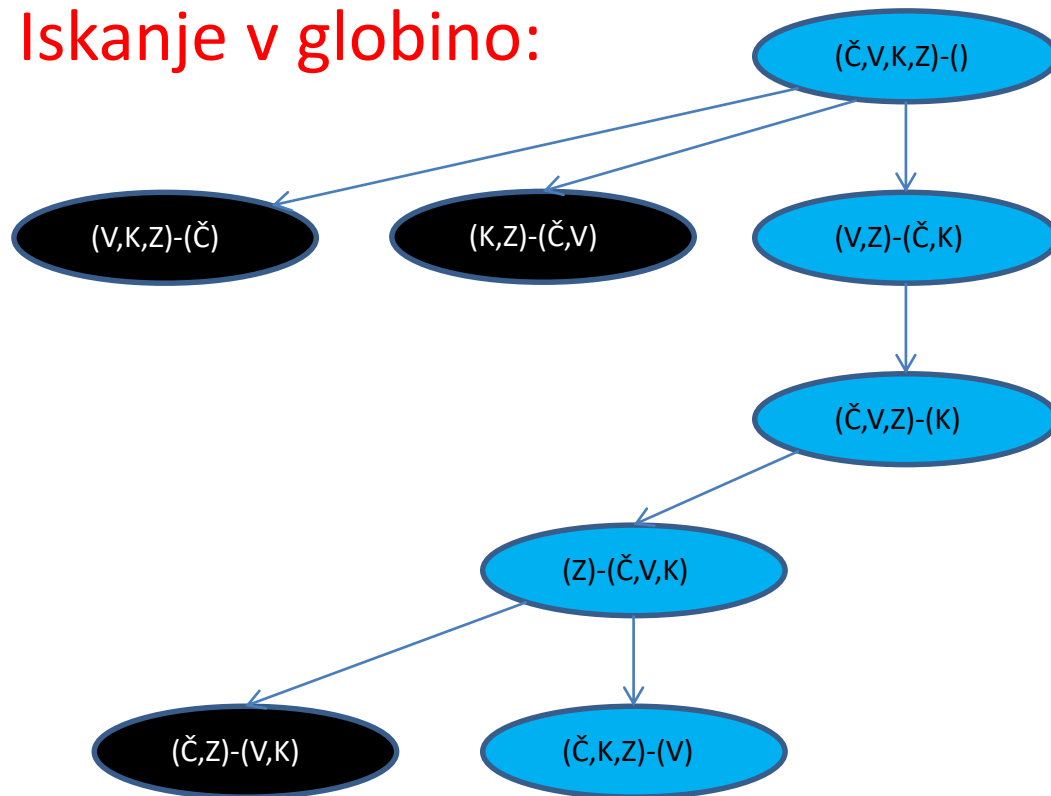
Iskanje v globino:



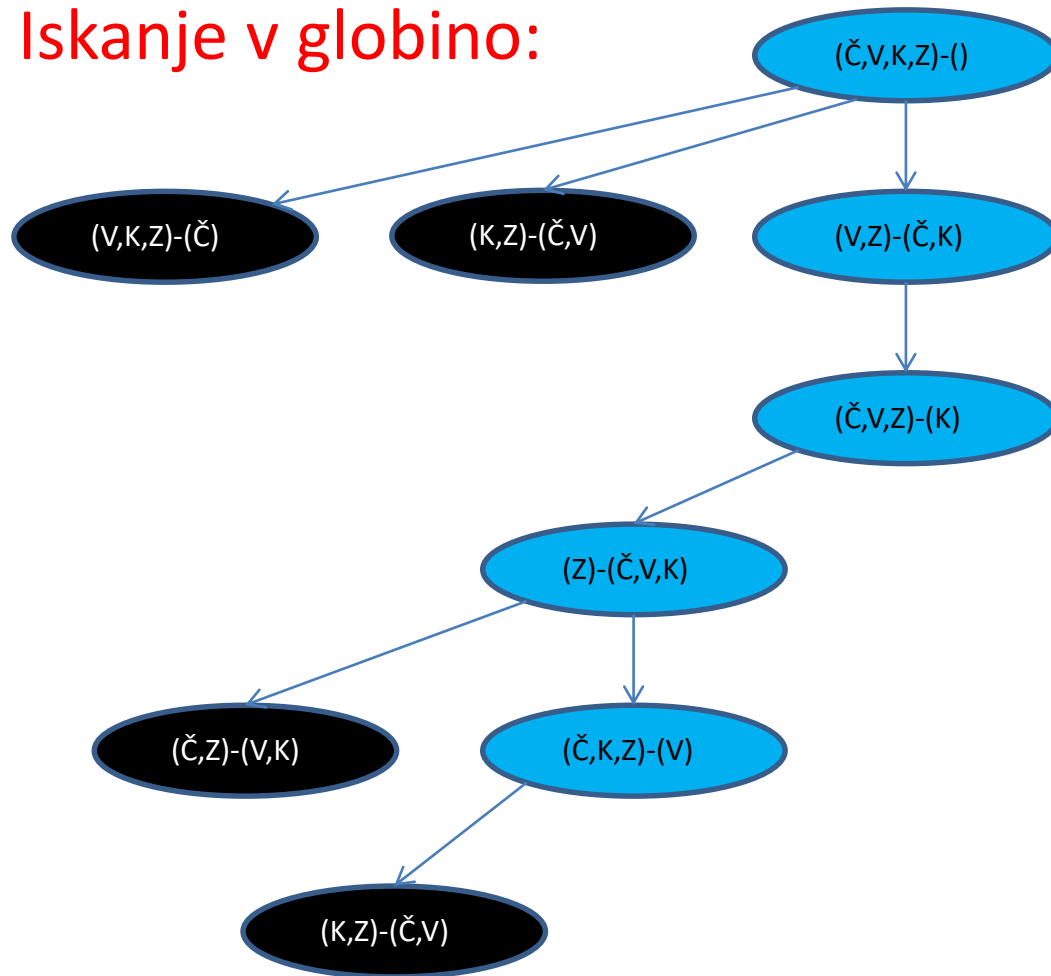
Iskanje v globino:



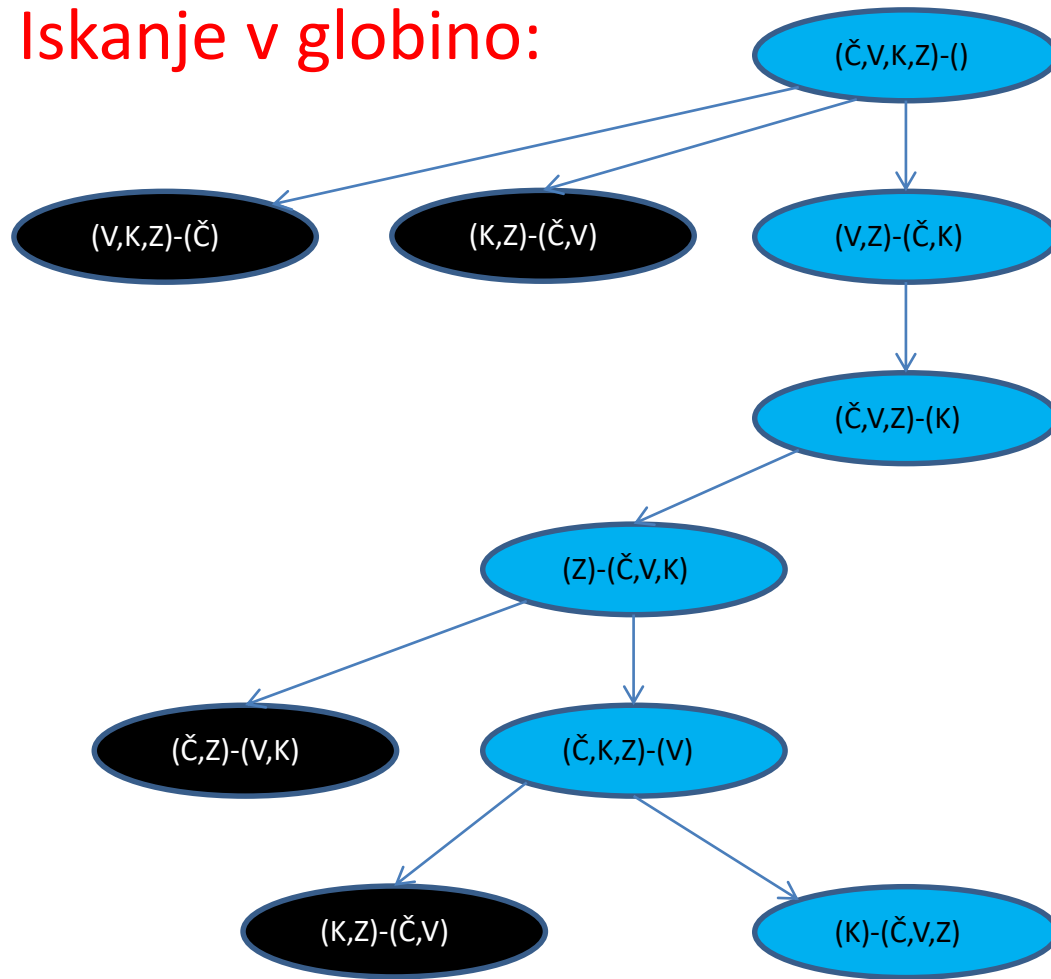
Iskanje v globino:



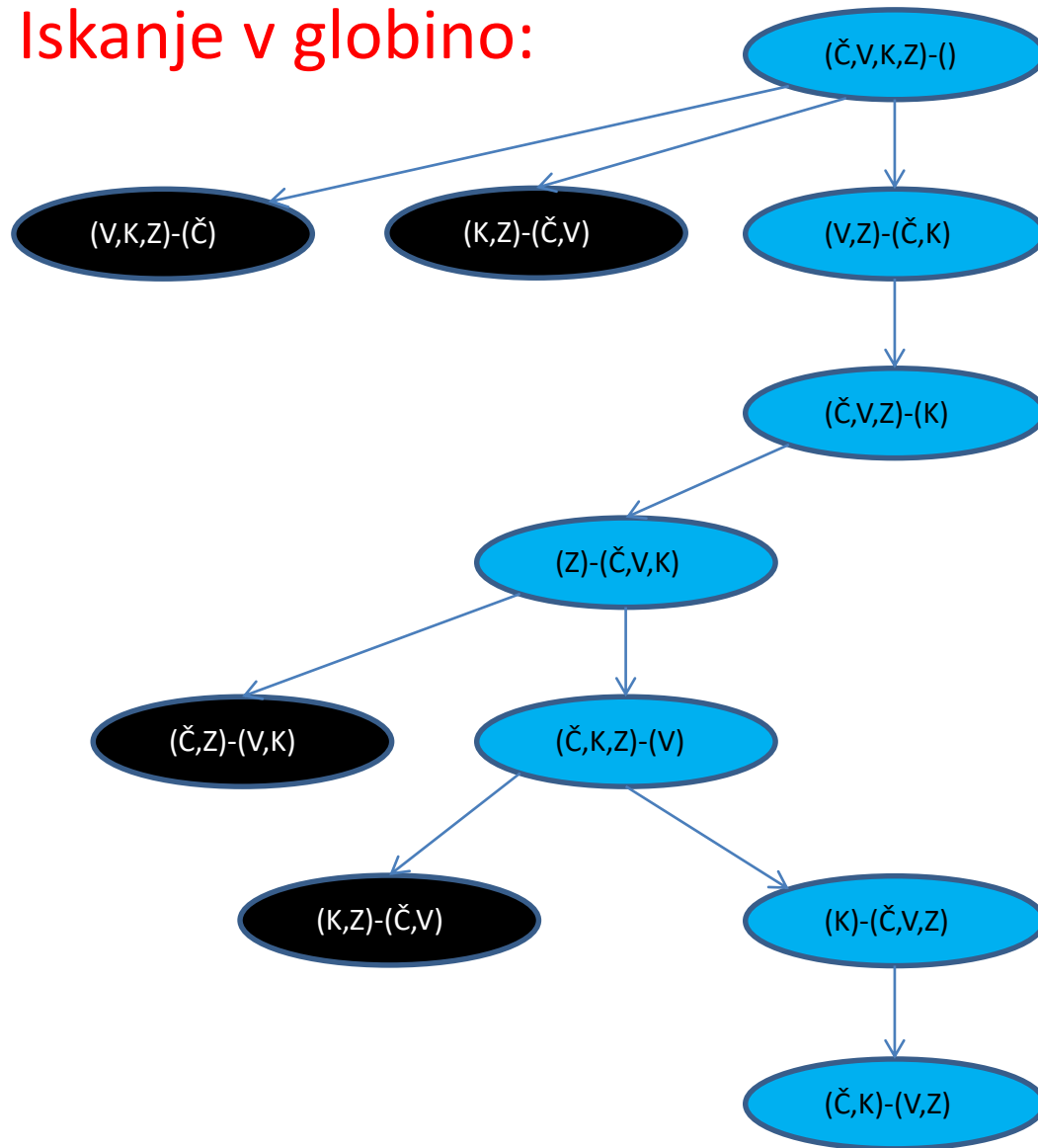
Iskanje v globino:



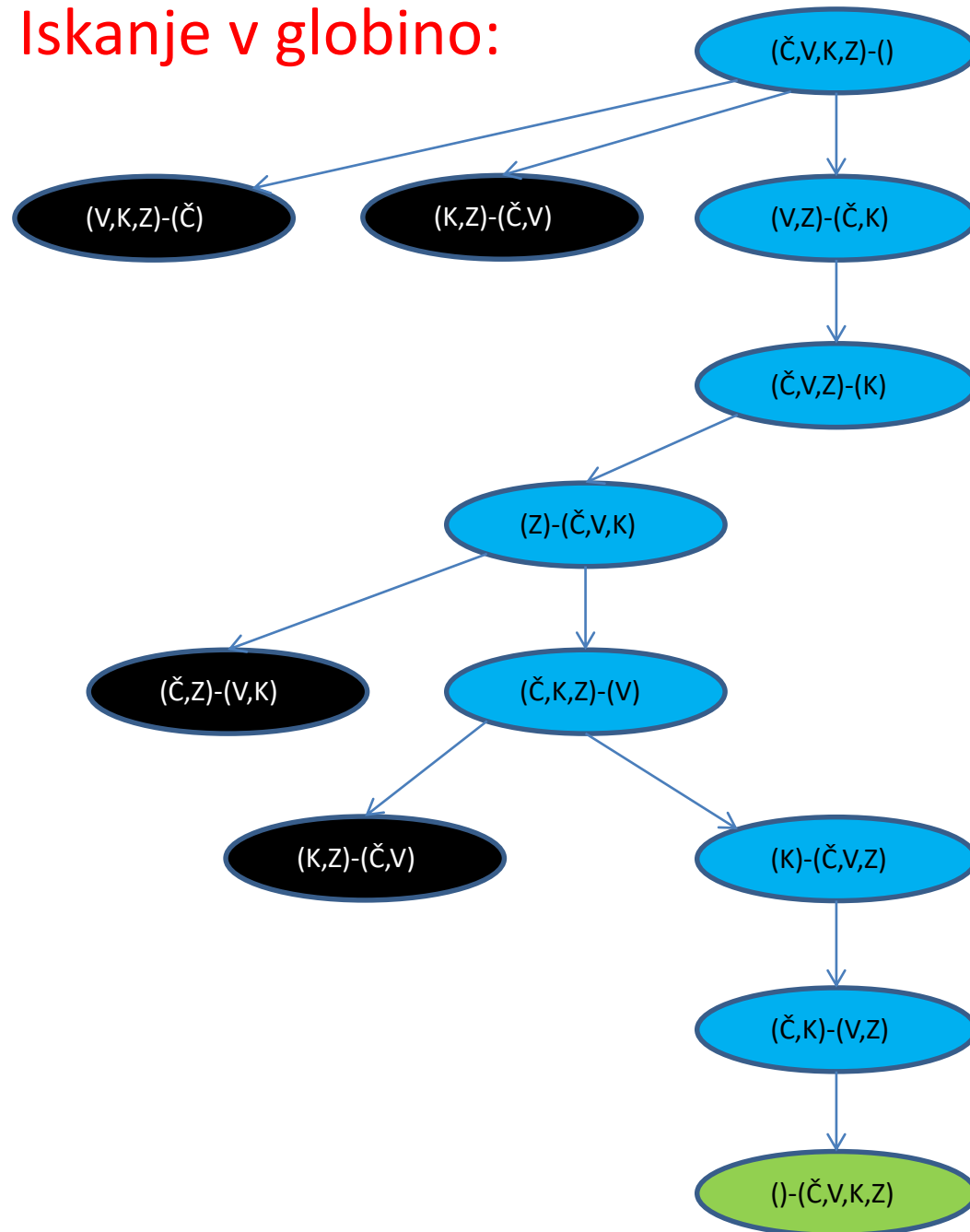
Iskanje v globino:



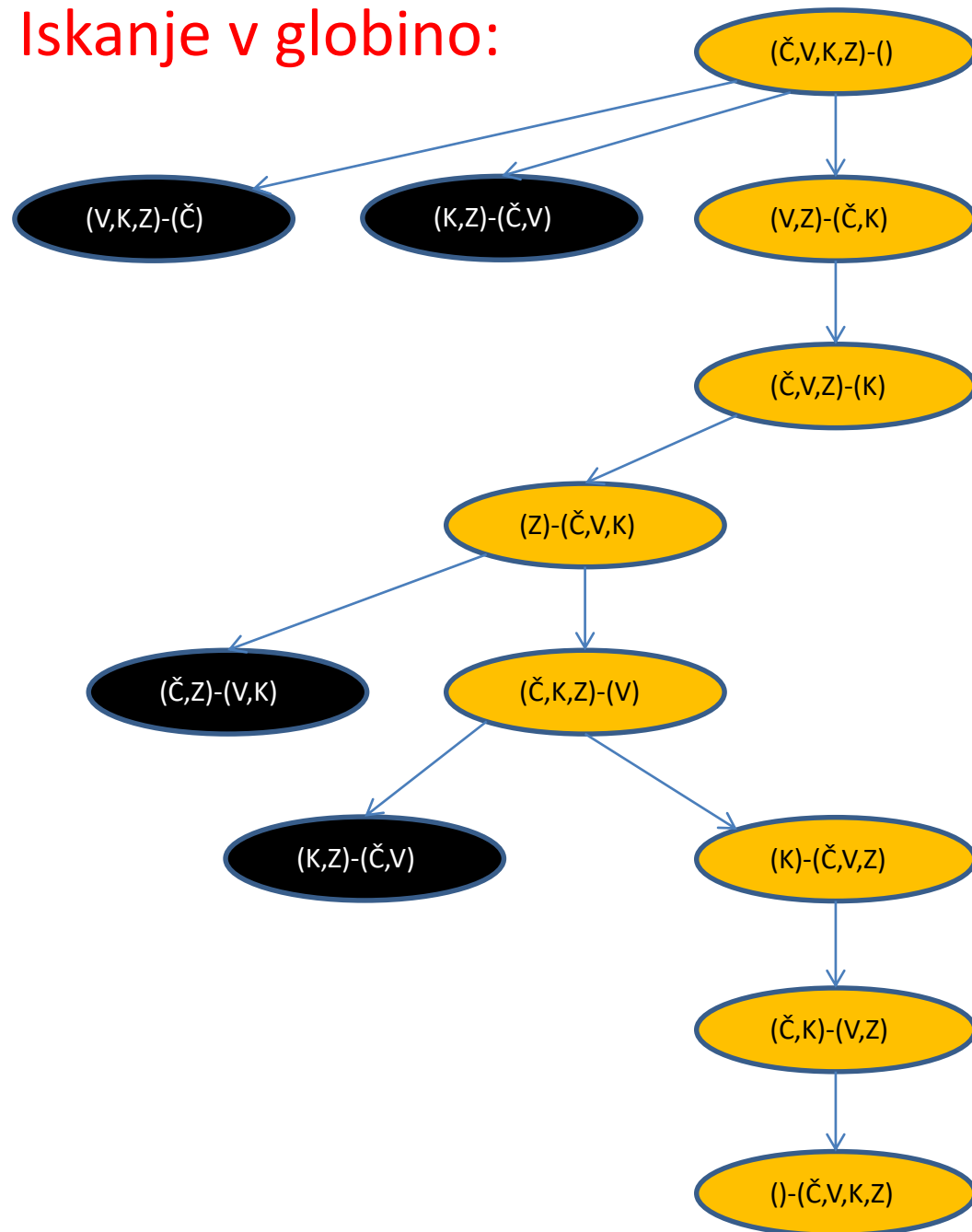
Iskanje v globino:



Iskanje v globino:

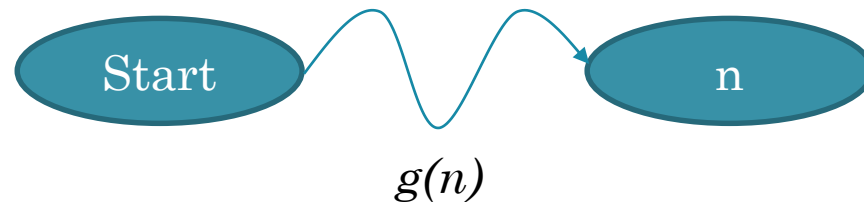


Iskanje v globino:



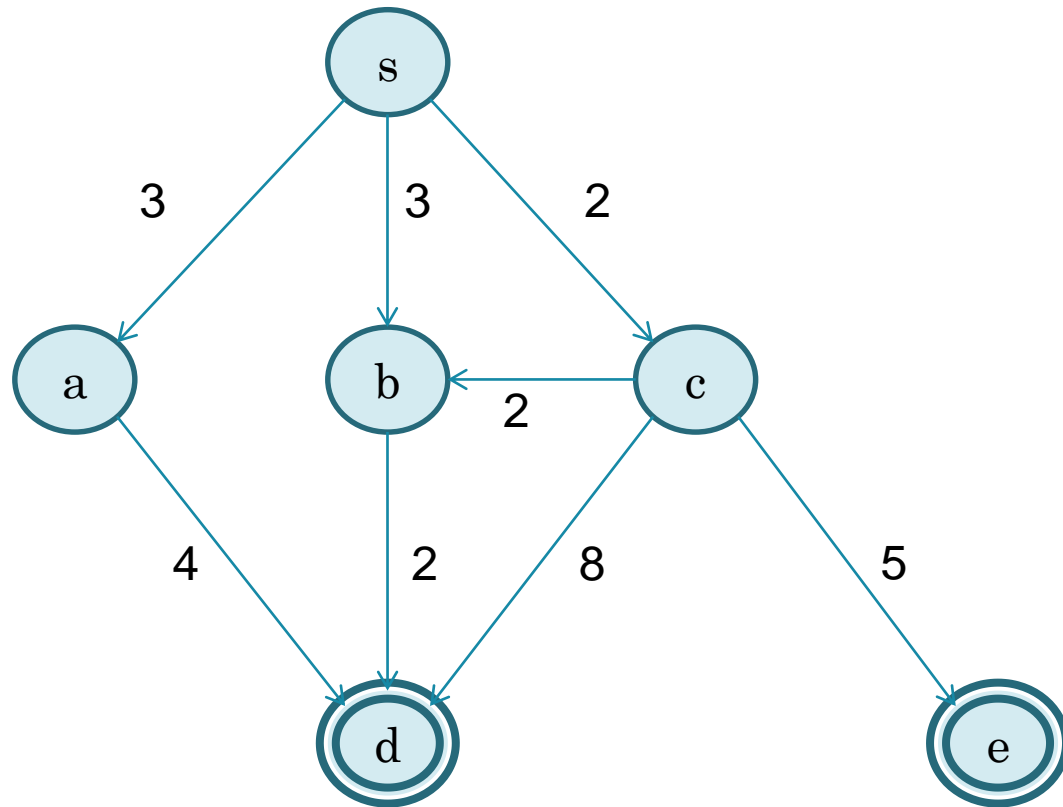
CENOVNO-OPTIMALNO PREISKOVANJE

- Posplošitev iskanja v širino (cene povezav so ≥ 1)
- Razvijamo vozlišče, ki ima najmanjšo skupno ceno dosedanje poti – $g(n)$
- Test, ali je vozlišče ciljno, opravimo šele, ko je vozlišče na vrsti za razvijanje

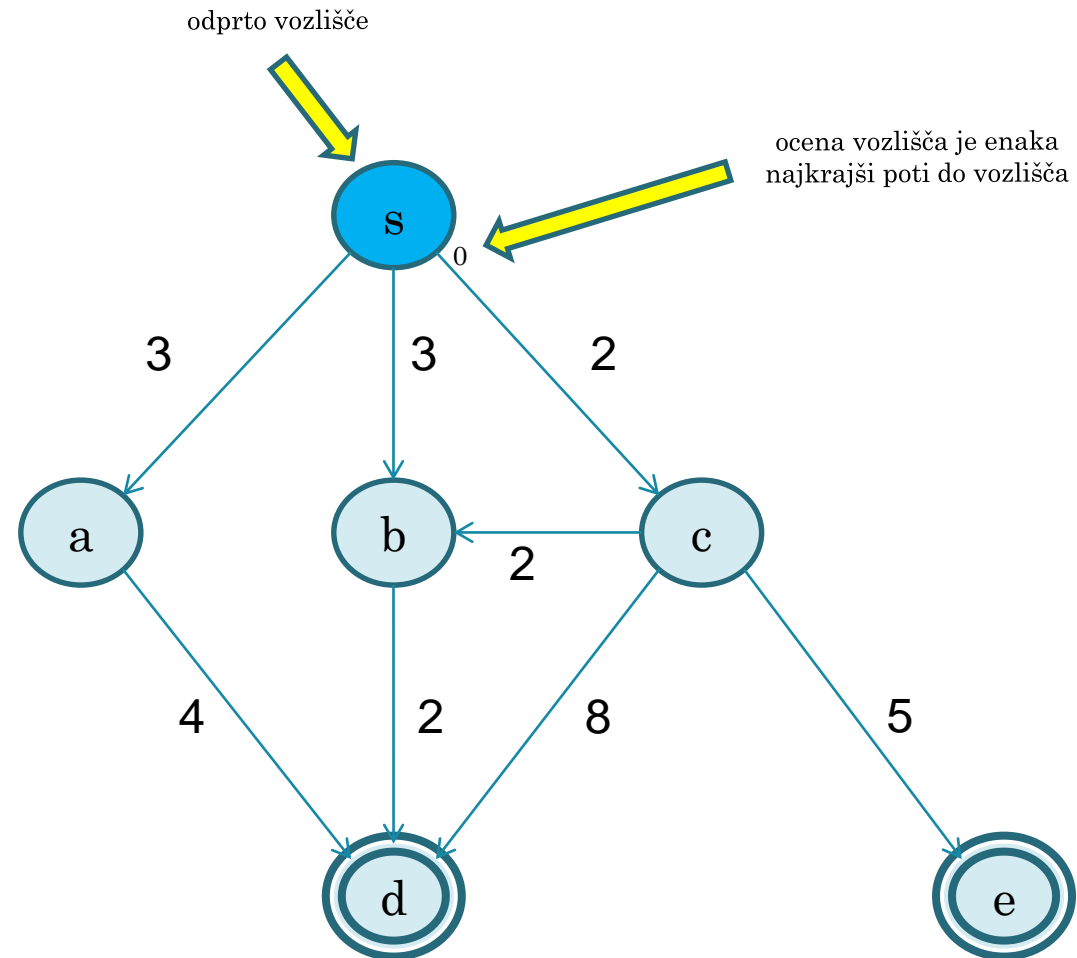


PRIMER – CENOVNO-OPTIMALNO ISKANJE (1/9)

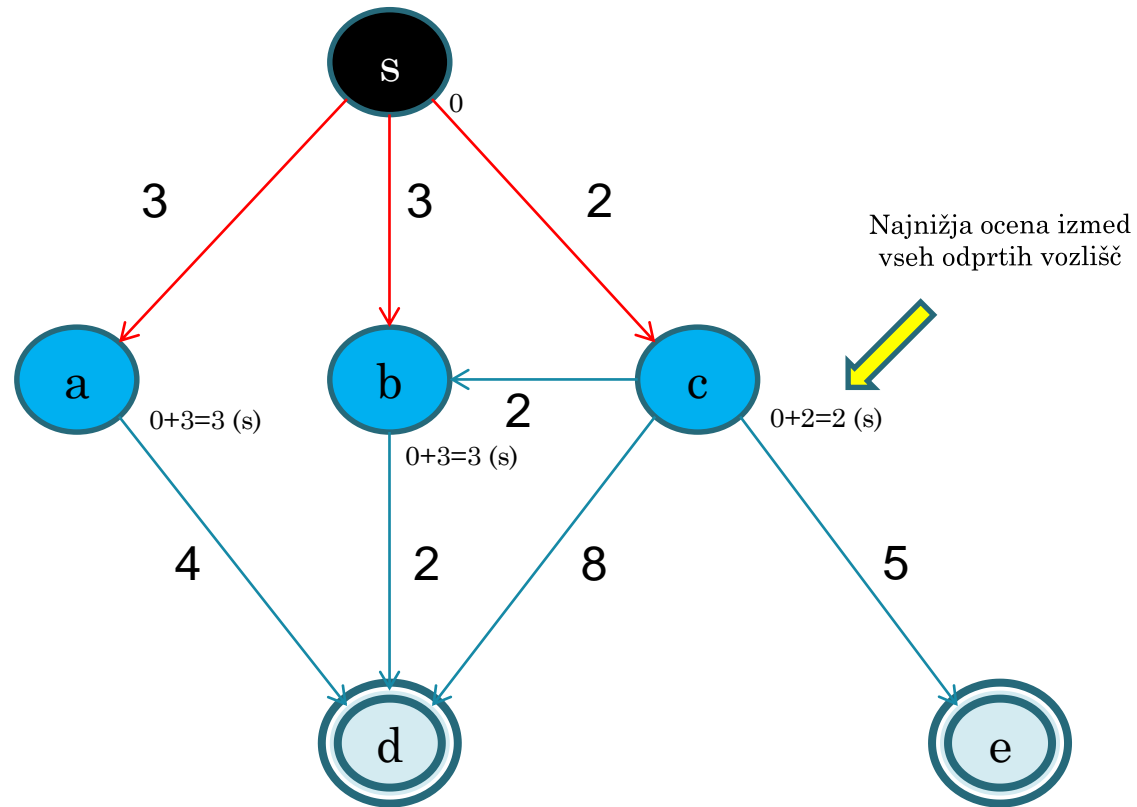
s – začetno vozlišče
d, e – končni vozlišči



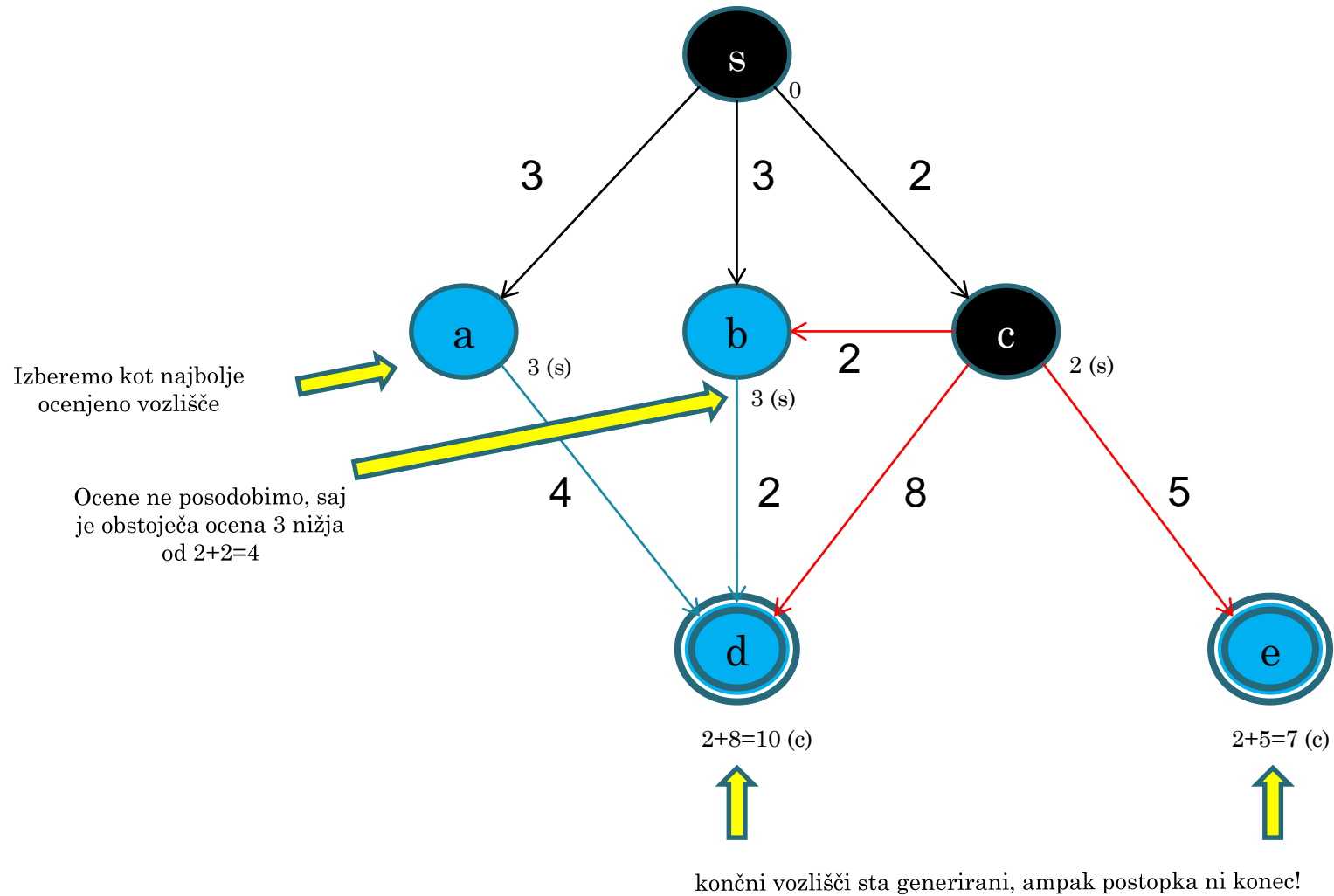
PRIMER – CENOVNO-OPTIMALNO ISKANJE (2/9)



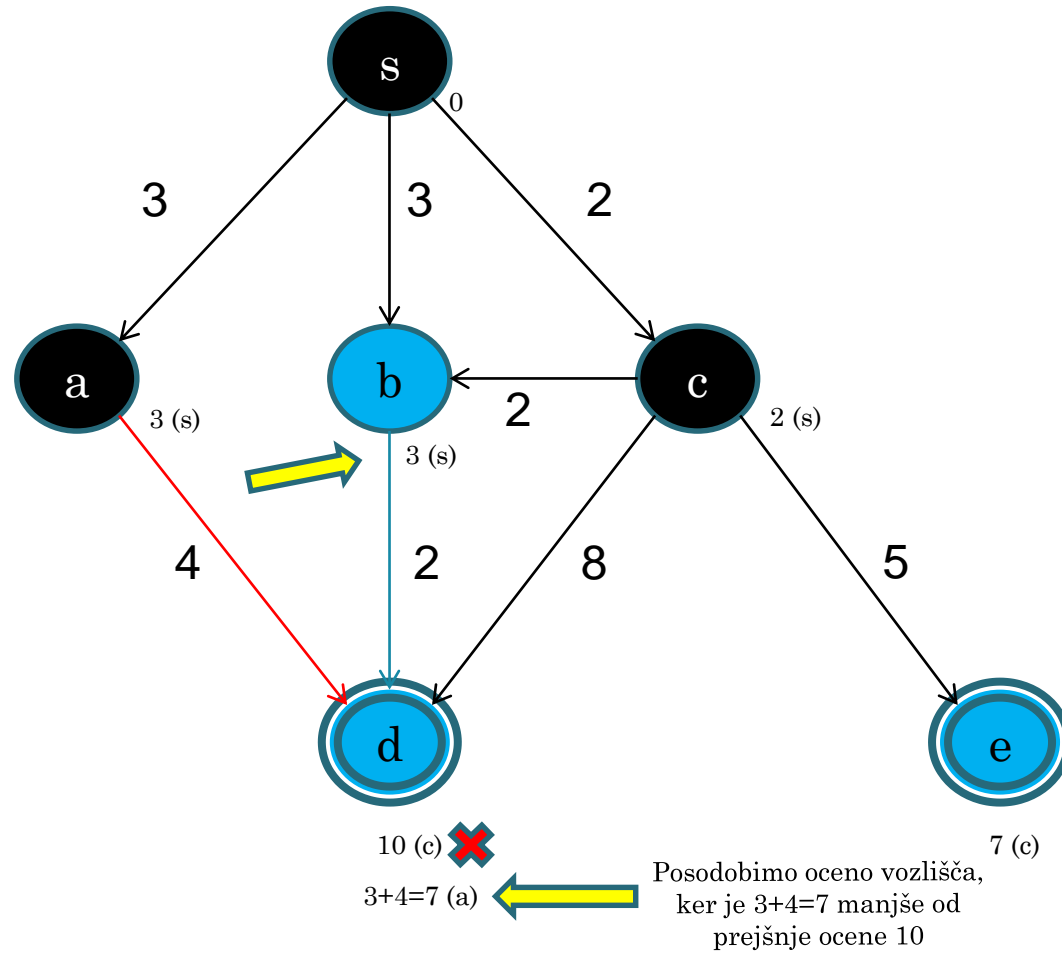
PRIMER – CENOVNO-OPTIMALNO ISKANJE (3/9)



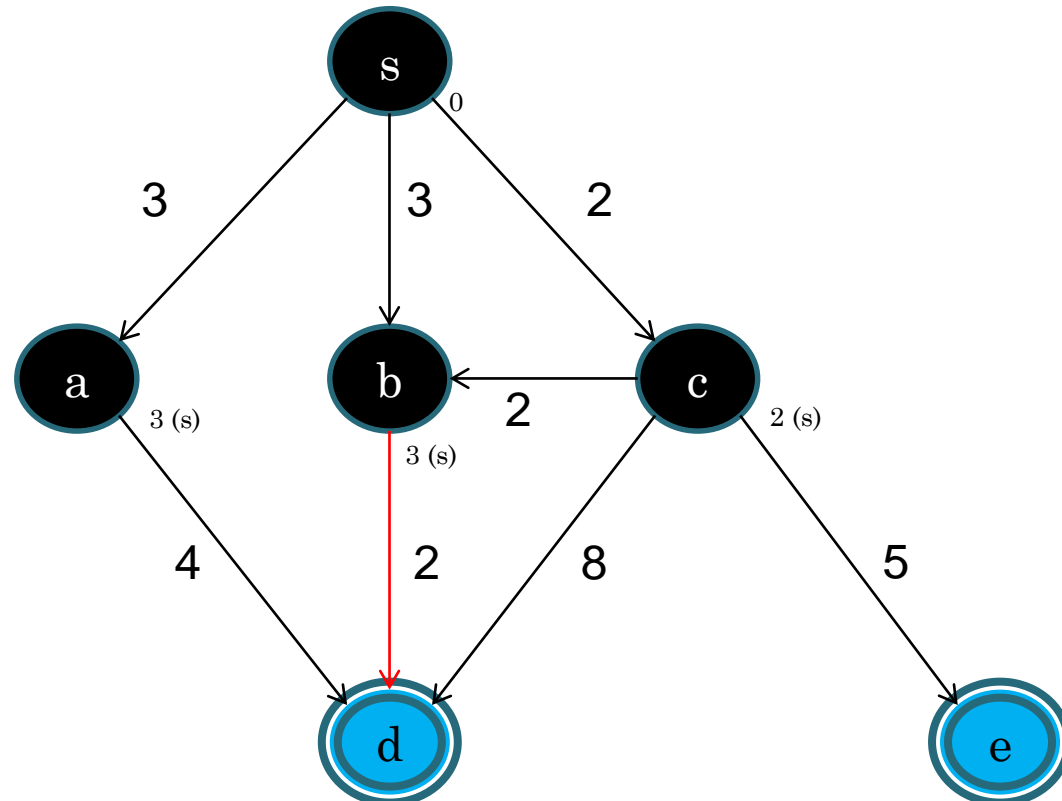
PRIMER – CENOVNO-OPTIMALNO ISKANJE (4/9)



PRIMER – CENOVNO-OPTIMALNO ISKANJE (5/9)



PRIMER – CENOVNO-OPTIMALNO ISKANJE (6/9)



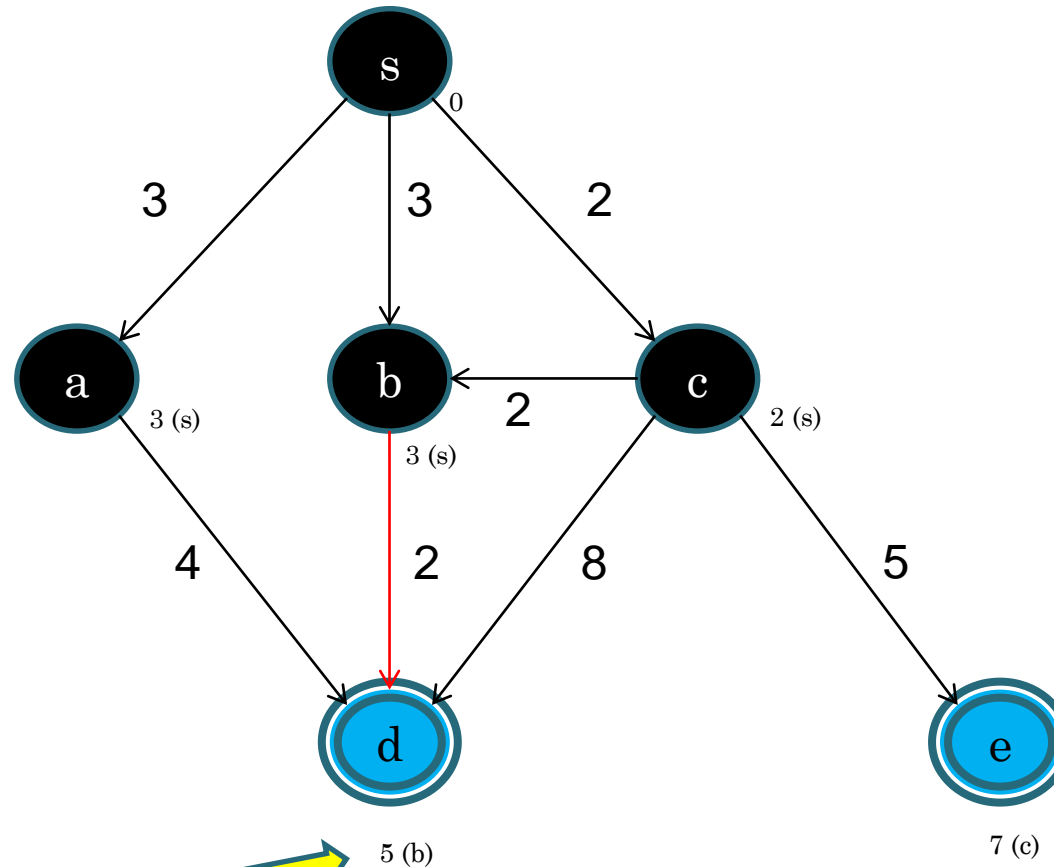
7 (a) ✘

3+2=5 (b)



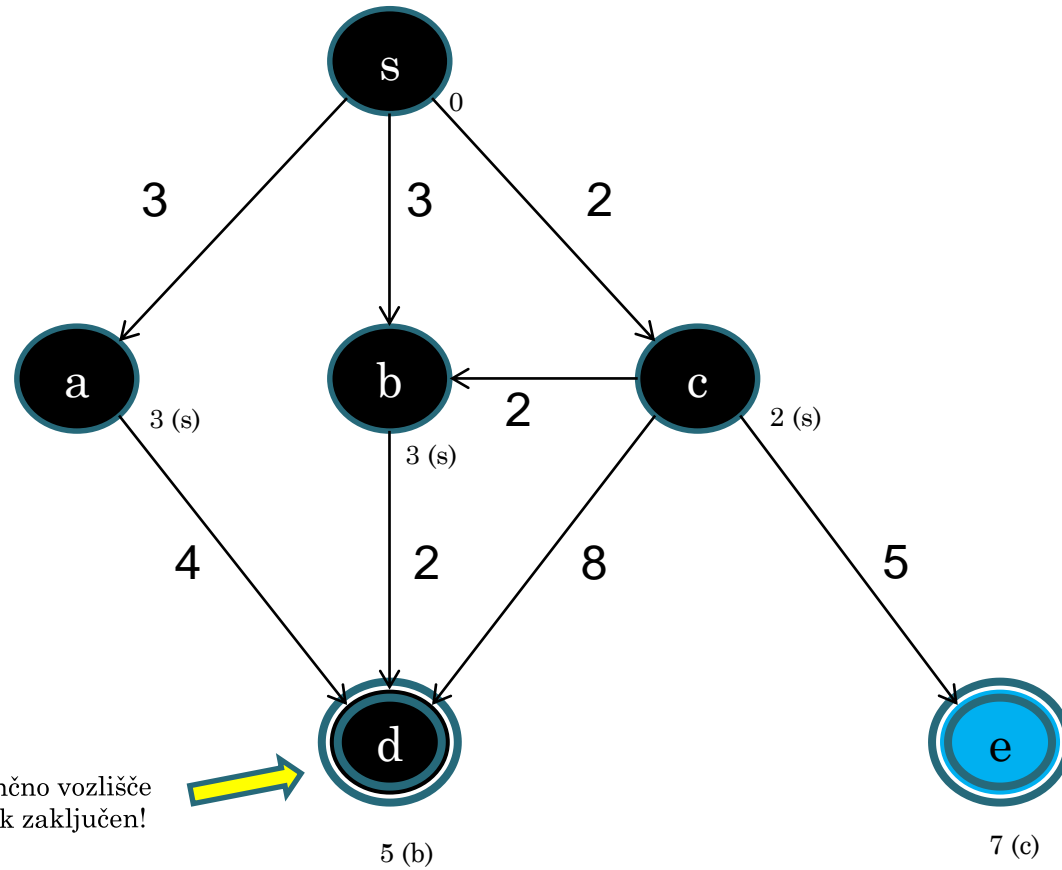
Ponovno posodobimo oceno vozlišča, ker je $3+2=5$ manjše od prejšnje ocene 7

PRIMER – CENOVNO-OPTIMALNO ISKANJE (7/9)



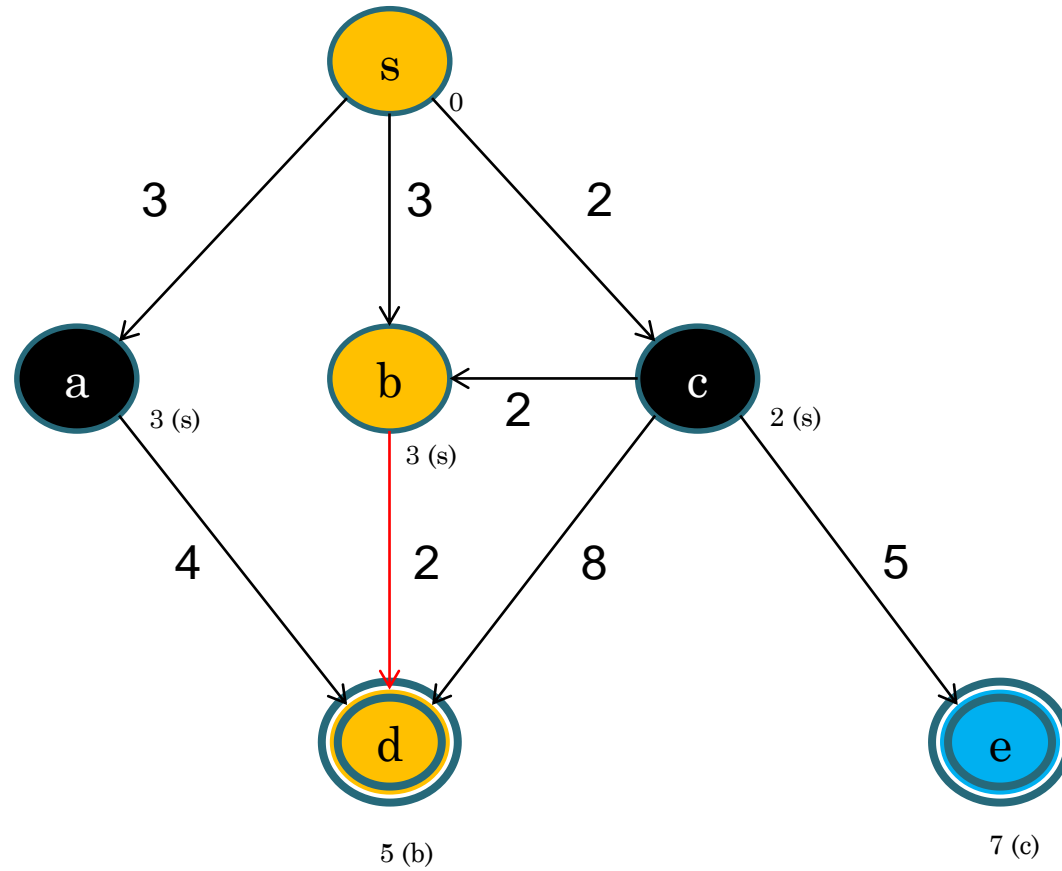
Najbolje ocenjeno vozlišče

PRIMER – CENOVNO-OPTIMALNO ISKANJE (8/9)



Razvili smo končno vozlišče
zato je postopek zaključen!

PRIMER – CENOVNO-OPTIMALNO ISKANJE (9/9)



HEVRISTIČNO PREISKOVANJE

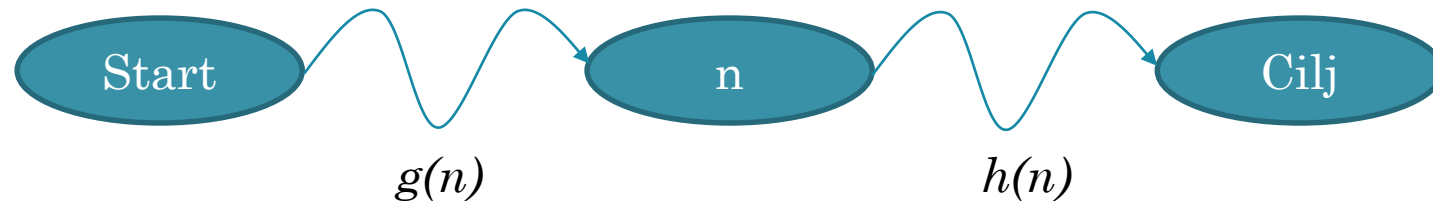
Če je prostor stanj velik, nam grozi kombinatorična eksplozija.

Moramo se zadovoljiti s preiskovanjem skromne podmnožice celotnega prostora stanj.

Uporabljamo hevristične ocene za omejevanje in usmerjanje iskanja v smeri najbolj obetavnega vozlišča.

Kot hevristična cenilka služi funkcija $f(n)$, ki ocenjuje “težavnost” vozlišča n .

HEVRISTIČNO PREISKOVANJE



$g(n)$ je cena najboljše poti od začetnega vozlišča do vozlišča n

$h(n)$ je ocena cene optimalne poti od vozlišča n do končnega vozlišča

Požrešno usmerjeno iskanje: $f(n) = h(n)$
(Greedy best-first search)

A*: $f(n) = g(n) + h(n)$

DOPUSTNOST

- algoritem je dopusten (admissable), če vedno najde optimalno rešitev, če ta obstaja
- predpostavimo, da je za vsako vozlišče n v prostoru stanj $h^*(n)$ cena optimalne poti od n do najbližjega končnega vozlišča
- algoritem A^* , ki uporablja hevristično funkcijo $h(n)$, tako da je za vsak n

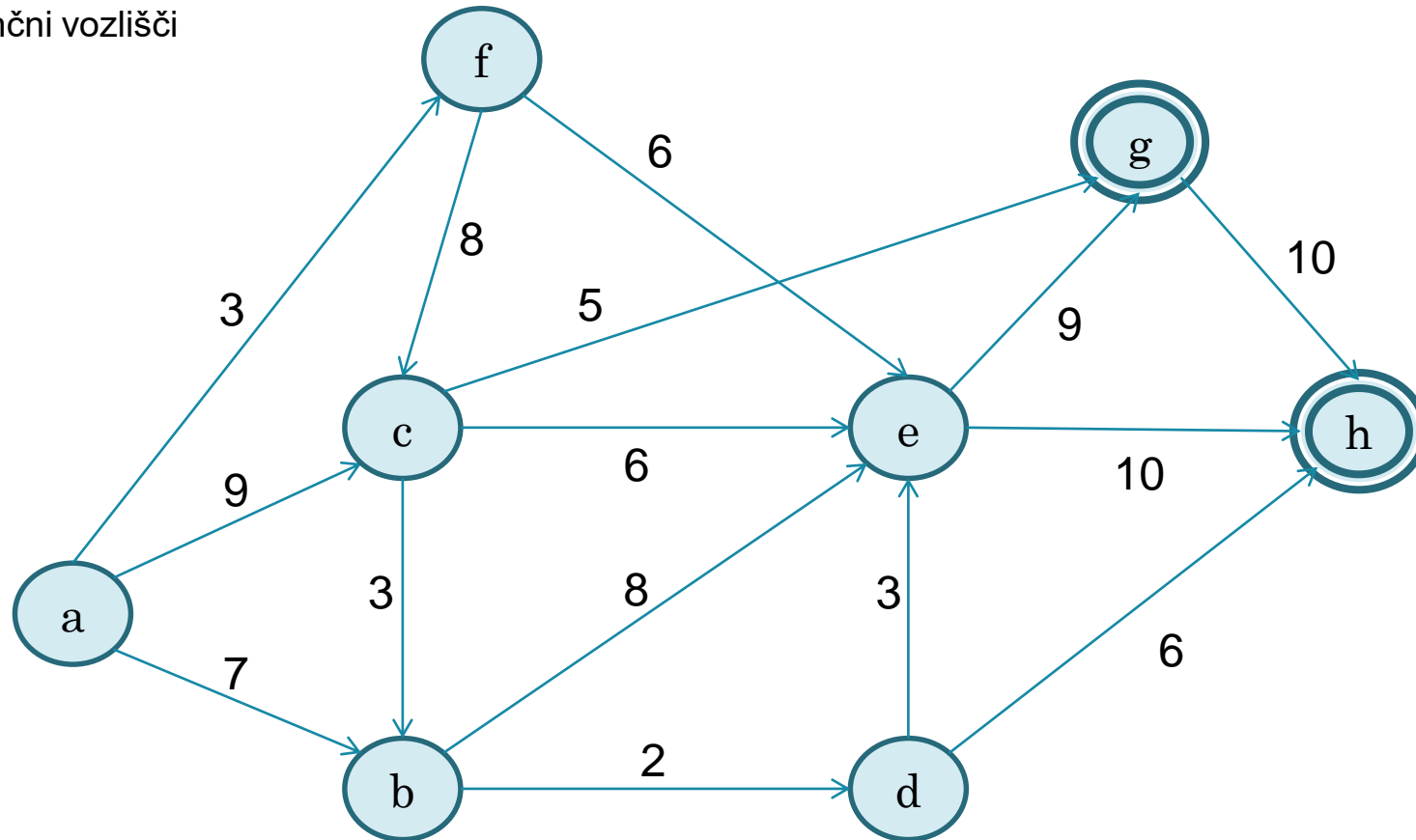
$$h(n) \leq h^*(n)$$

je dopusten.

- vsaka hevristična funkcija $h(n)$, ki ne precenjuje razdalje do cilja, je dopustna.
- trivialna, a neuporabna hevristična funkcija $h(n) = 0$ spremeni A^* v iskanje v širino

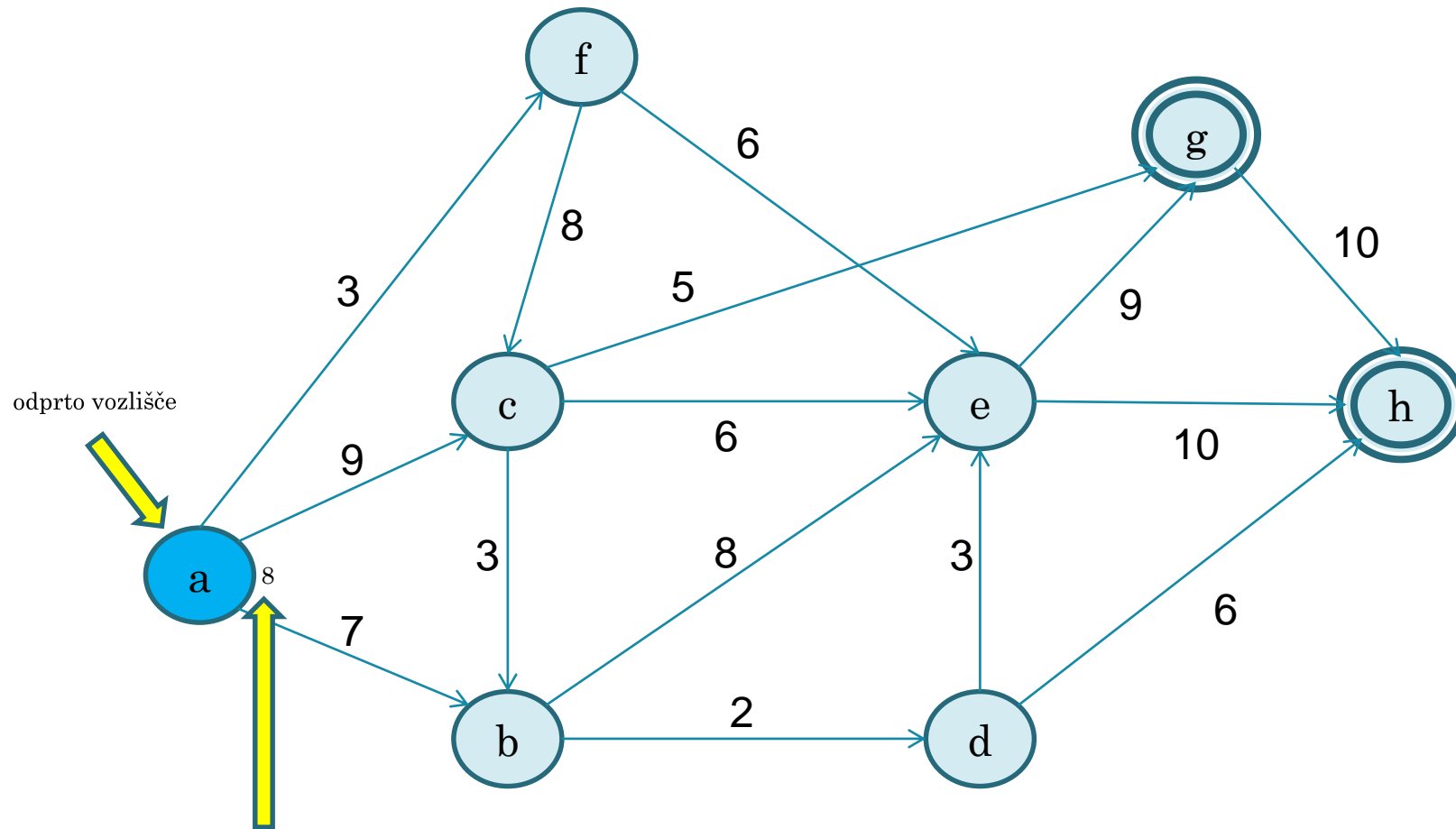
PRIMER – POŽREŠNO ISKANJE (1/7)

a – začetno vozlišče
g,h – končni vozlišči



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

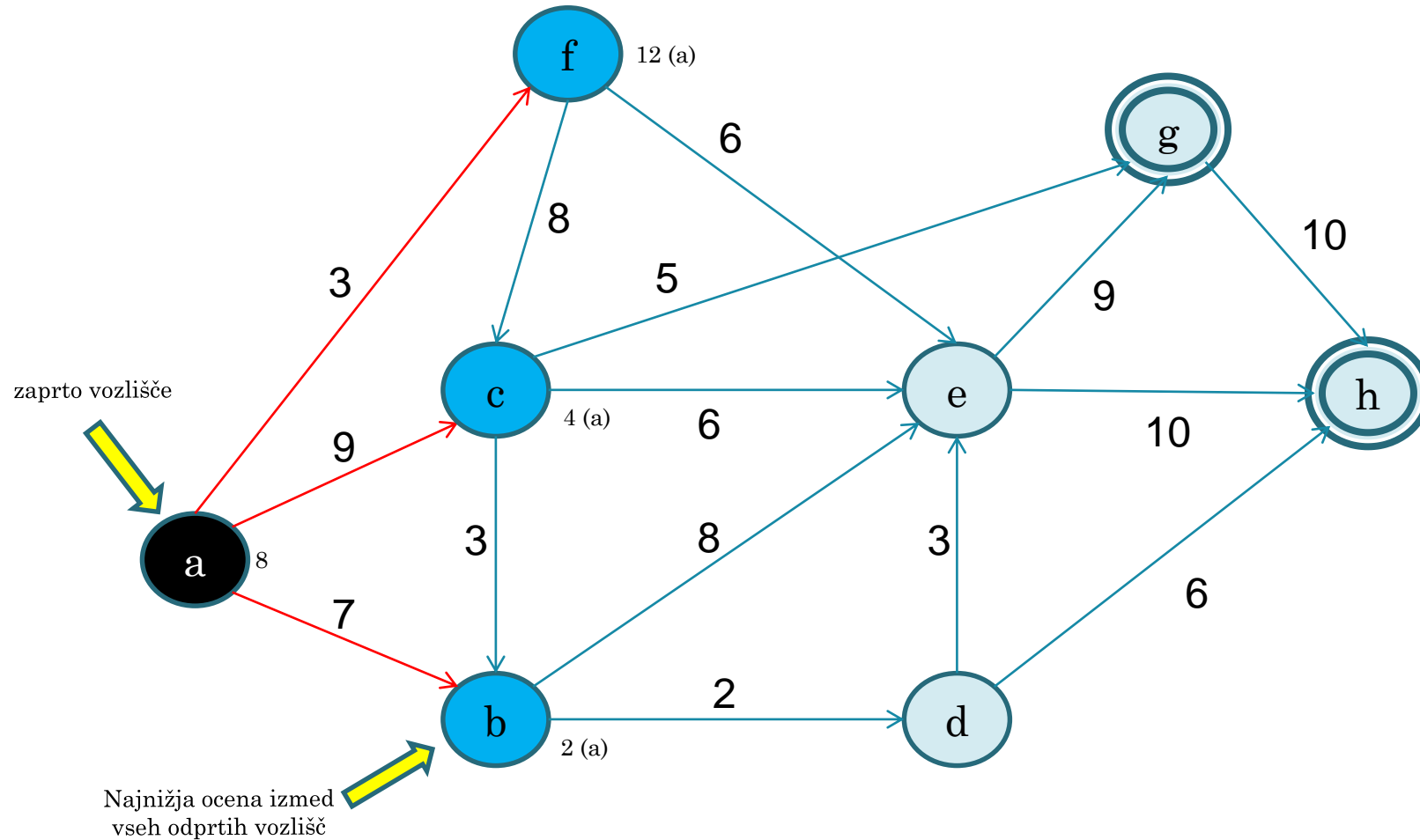
PRIMER – POŽREŠNO ISKANJE (2/7)



ocena vozlišča
je enaka
hevrstiki h

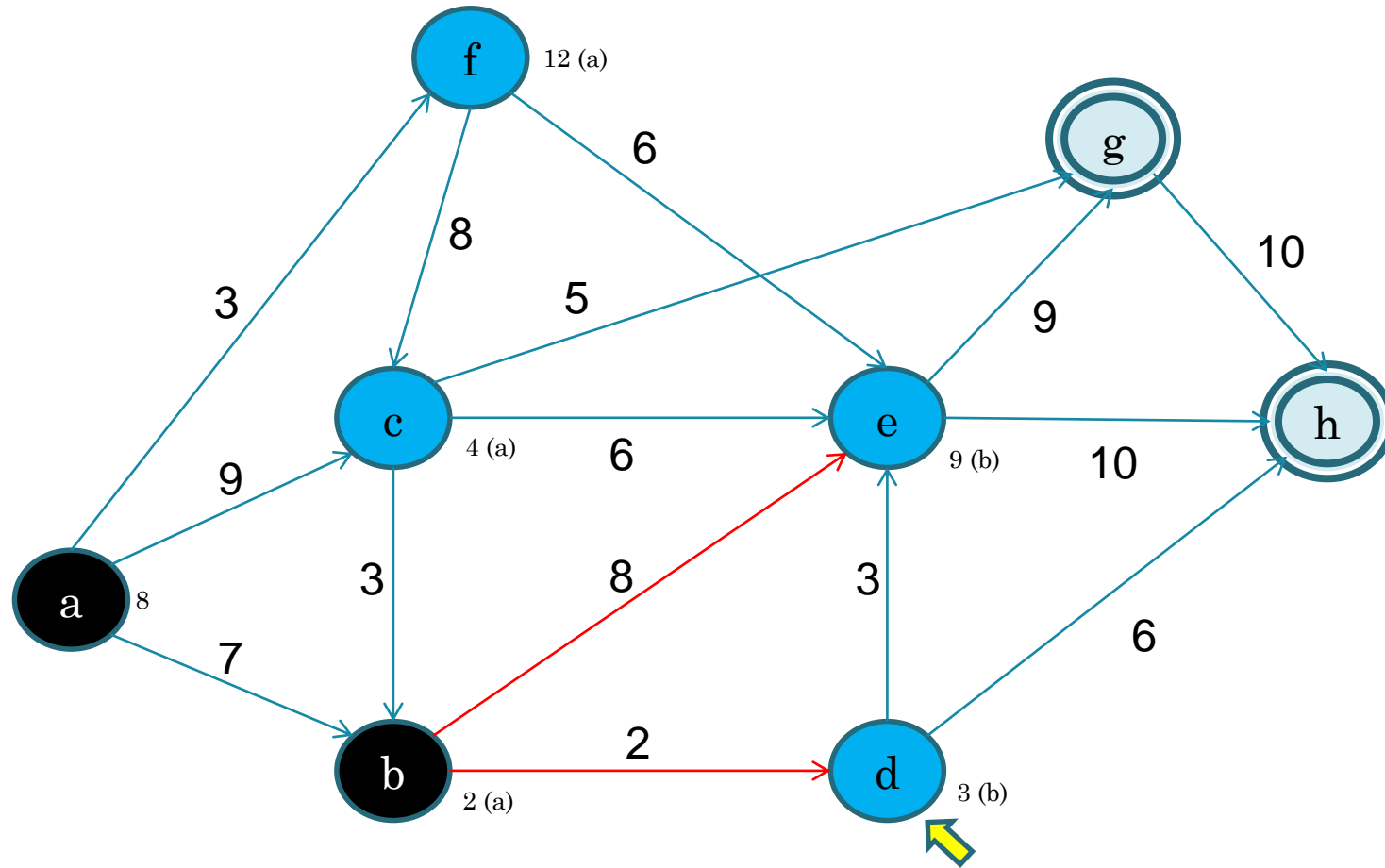
	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – POŽREŠNO ISKANJE (3/7)



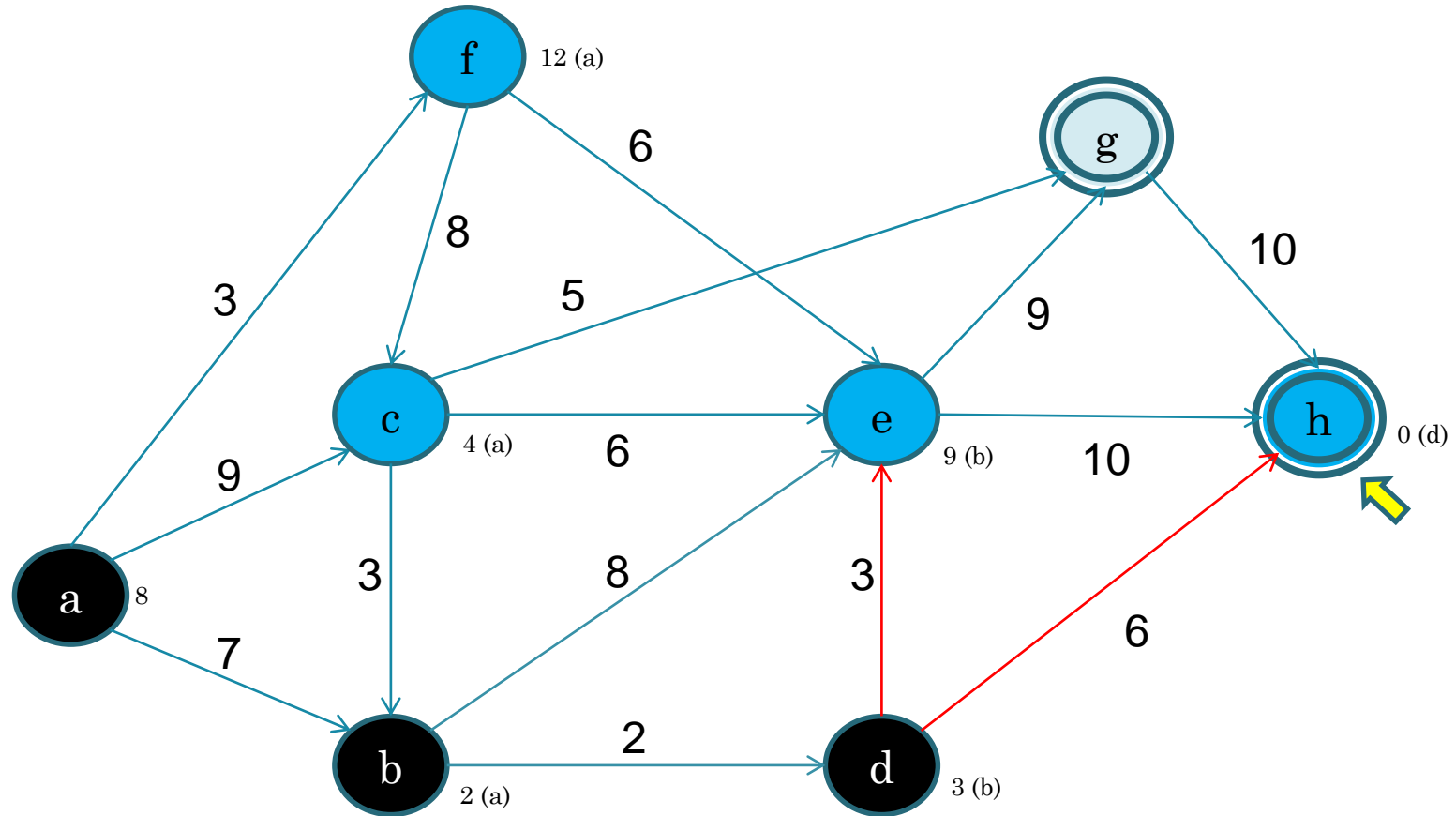
	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – POŽREŠNO ISKANJE (4/7)



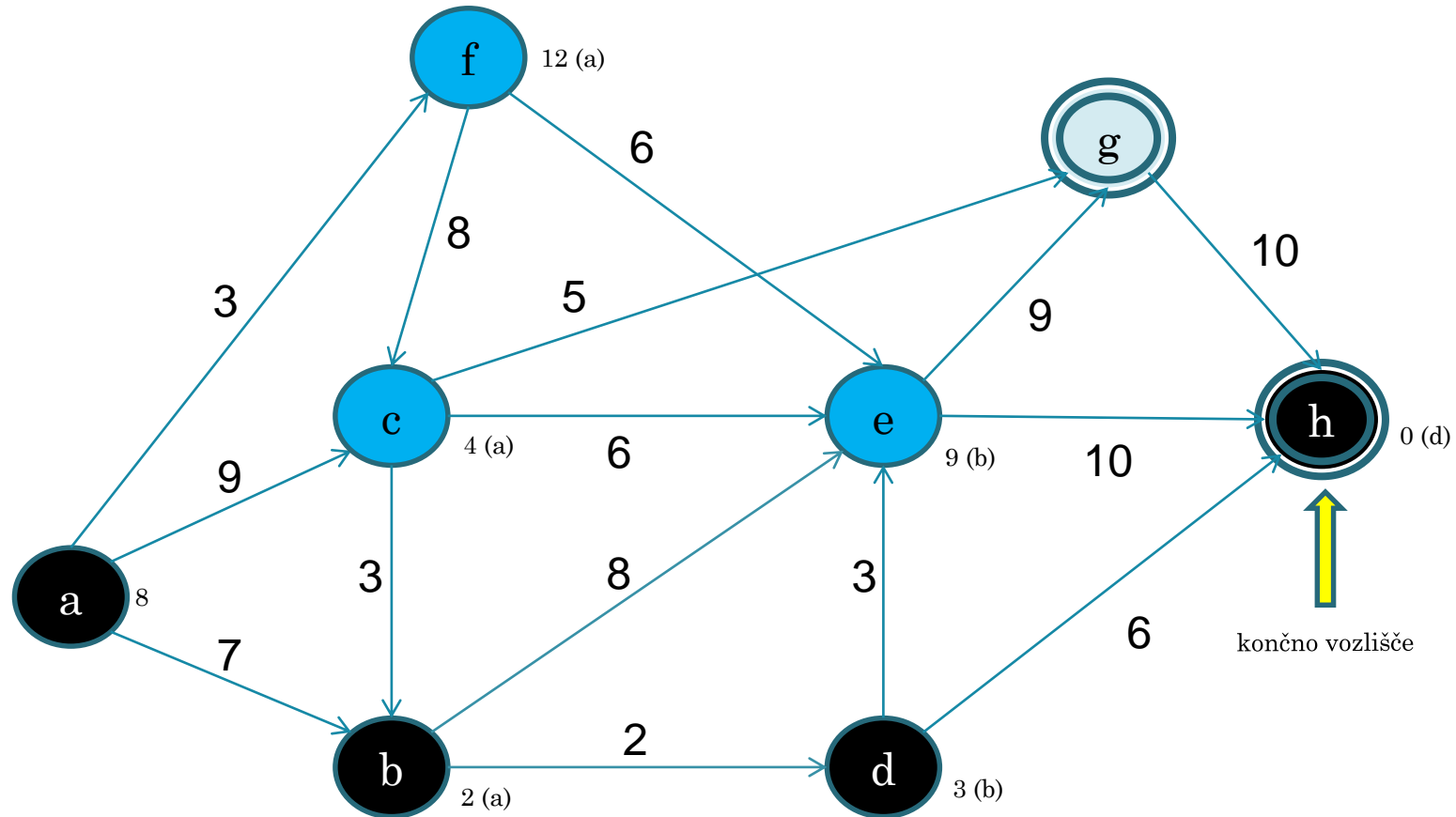
	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – POŽREŠNO ISKANJE (5/7)



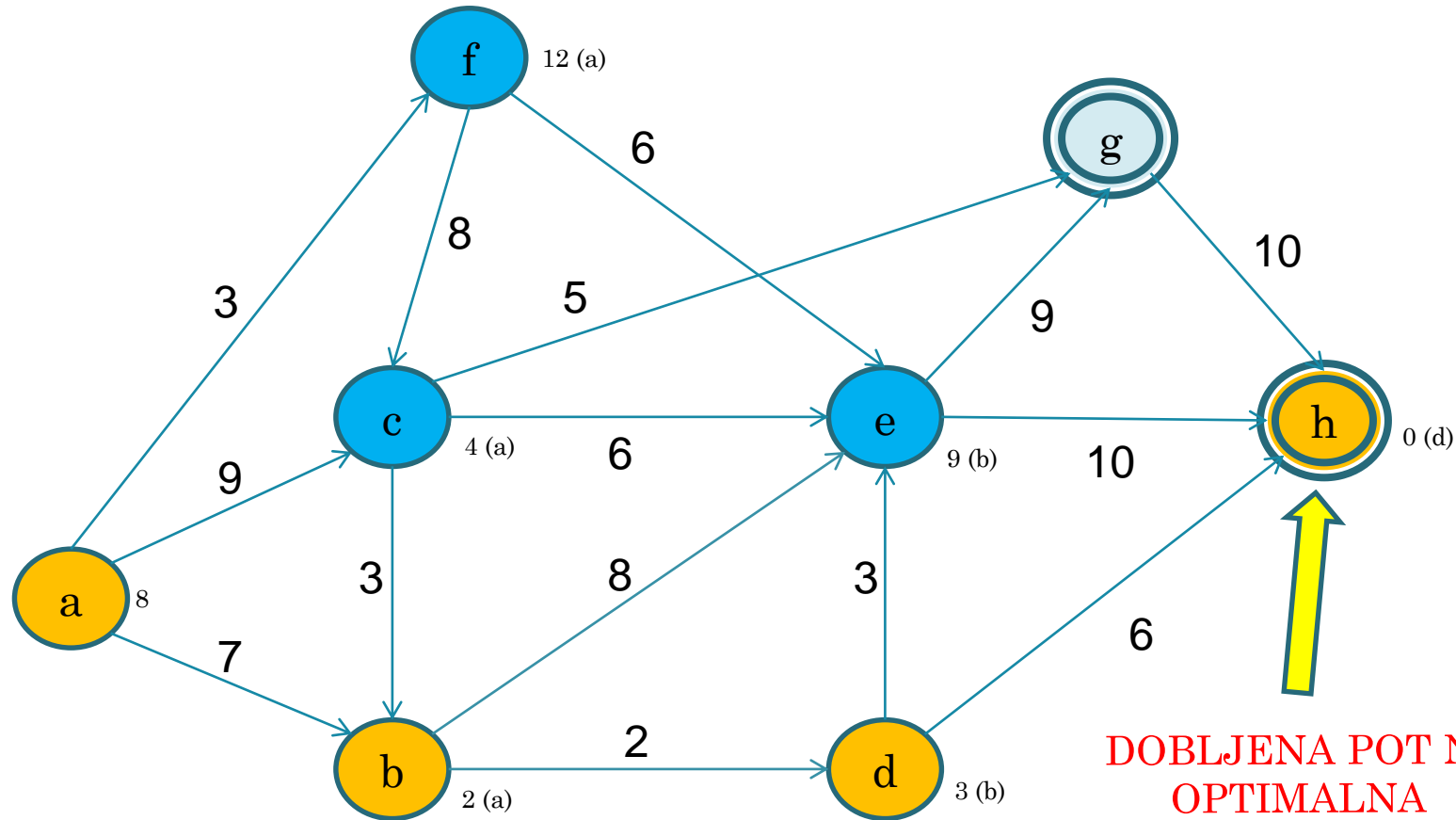
	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – POŽREŠNO ISKANJE (6/7)



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – POŽREŠNO ISKANJE (7/7)

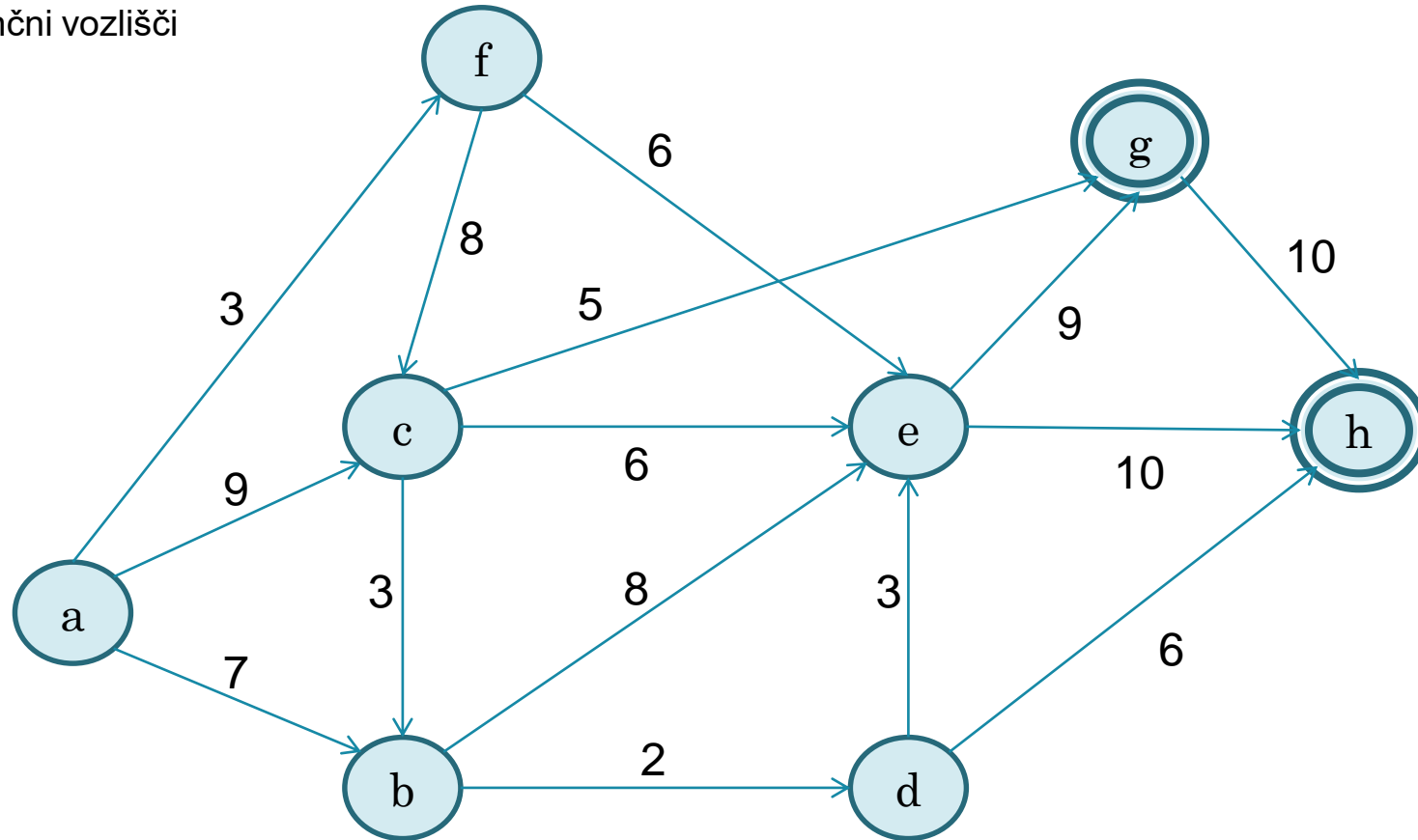


**DOBLJENA POT NI
OPTIMALNA
REŠITEV!**

	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

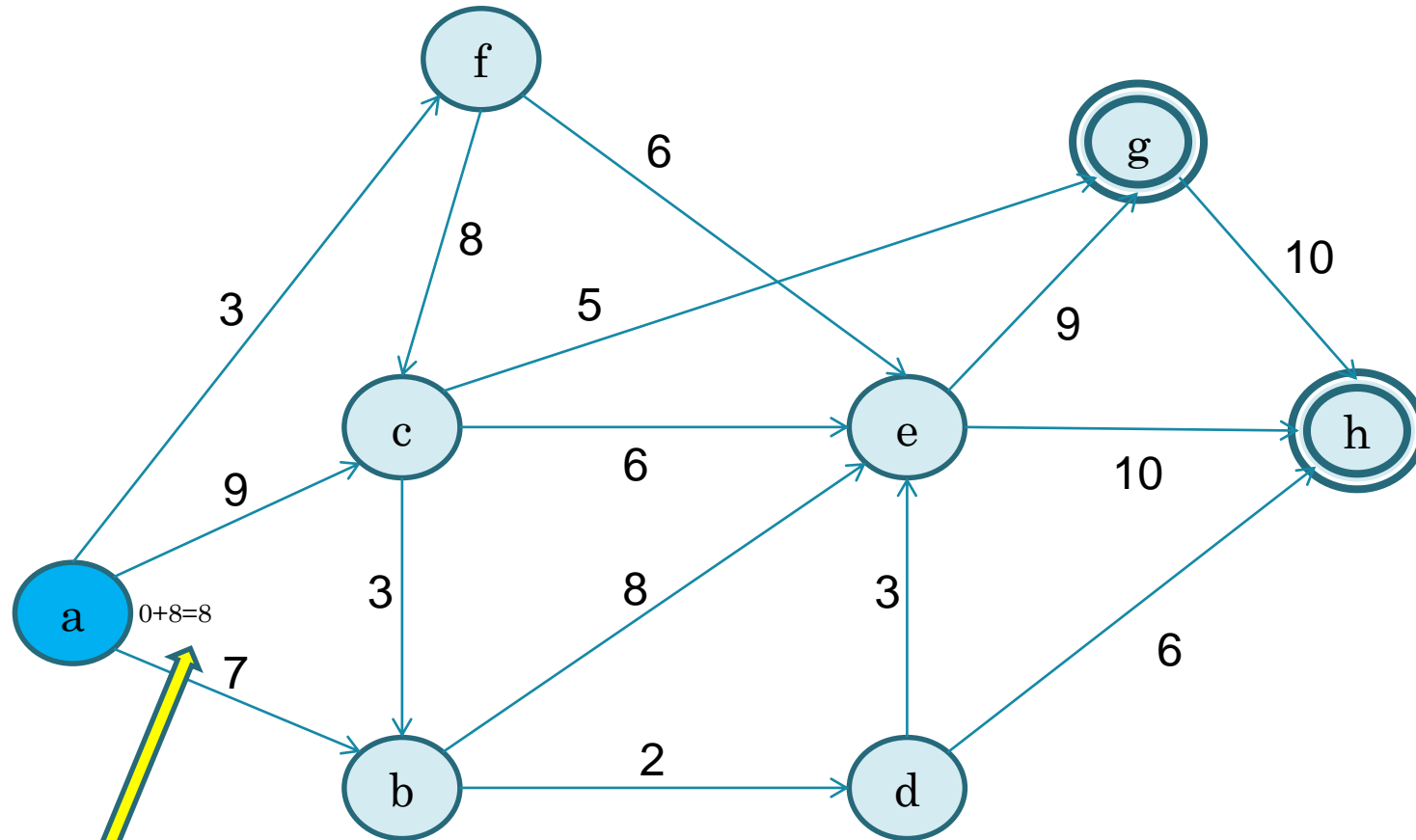
PRIMER – A* (1/8)

a – začetno vozlišče
g,h – končni vozlišči



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

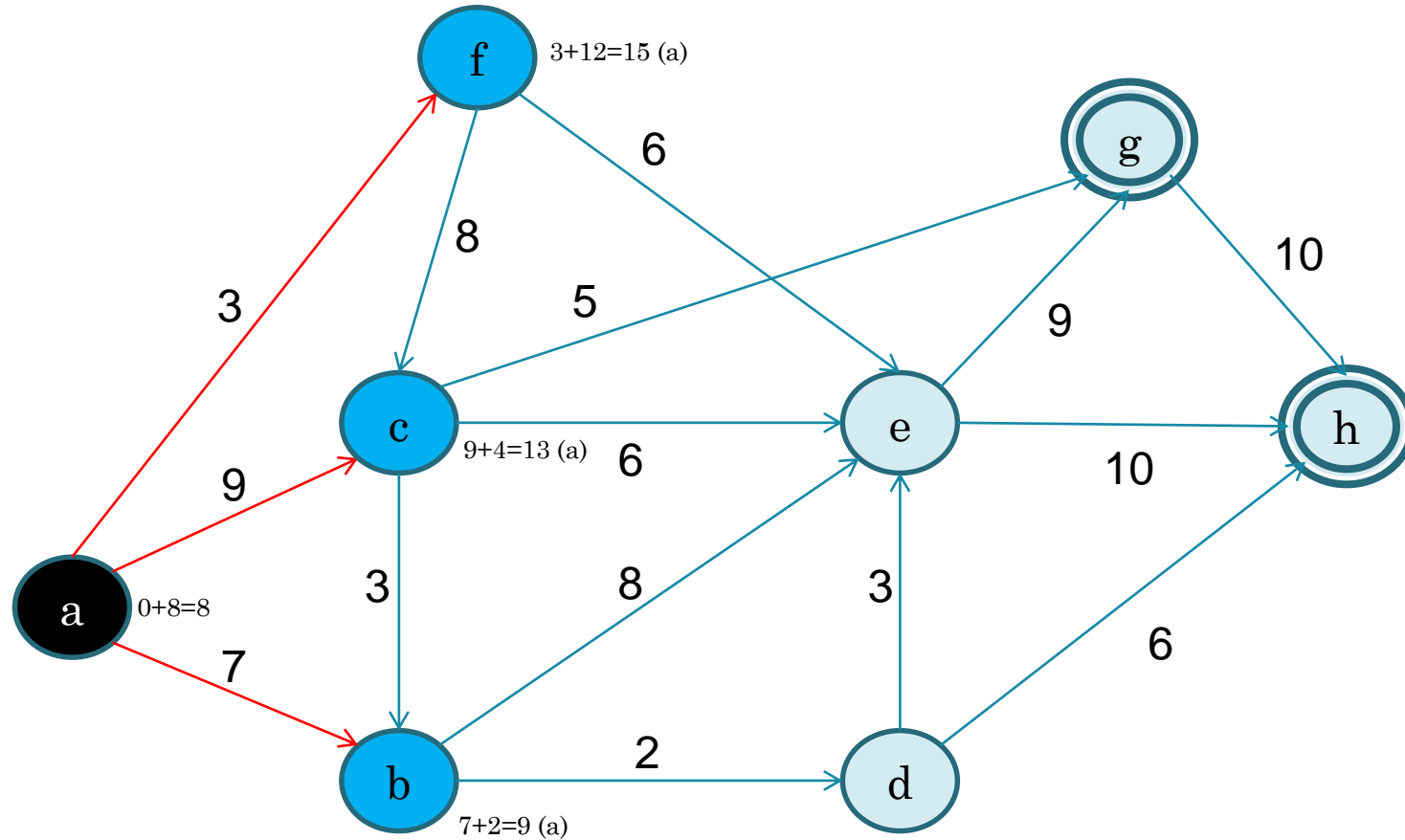
PRIMER – A* (2/8)



ocena vozlišča je sestavljena iz cene najboljše poti od začetnega vozlišča in heuristike h

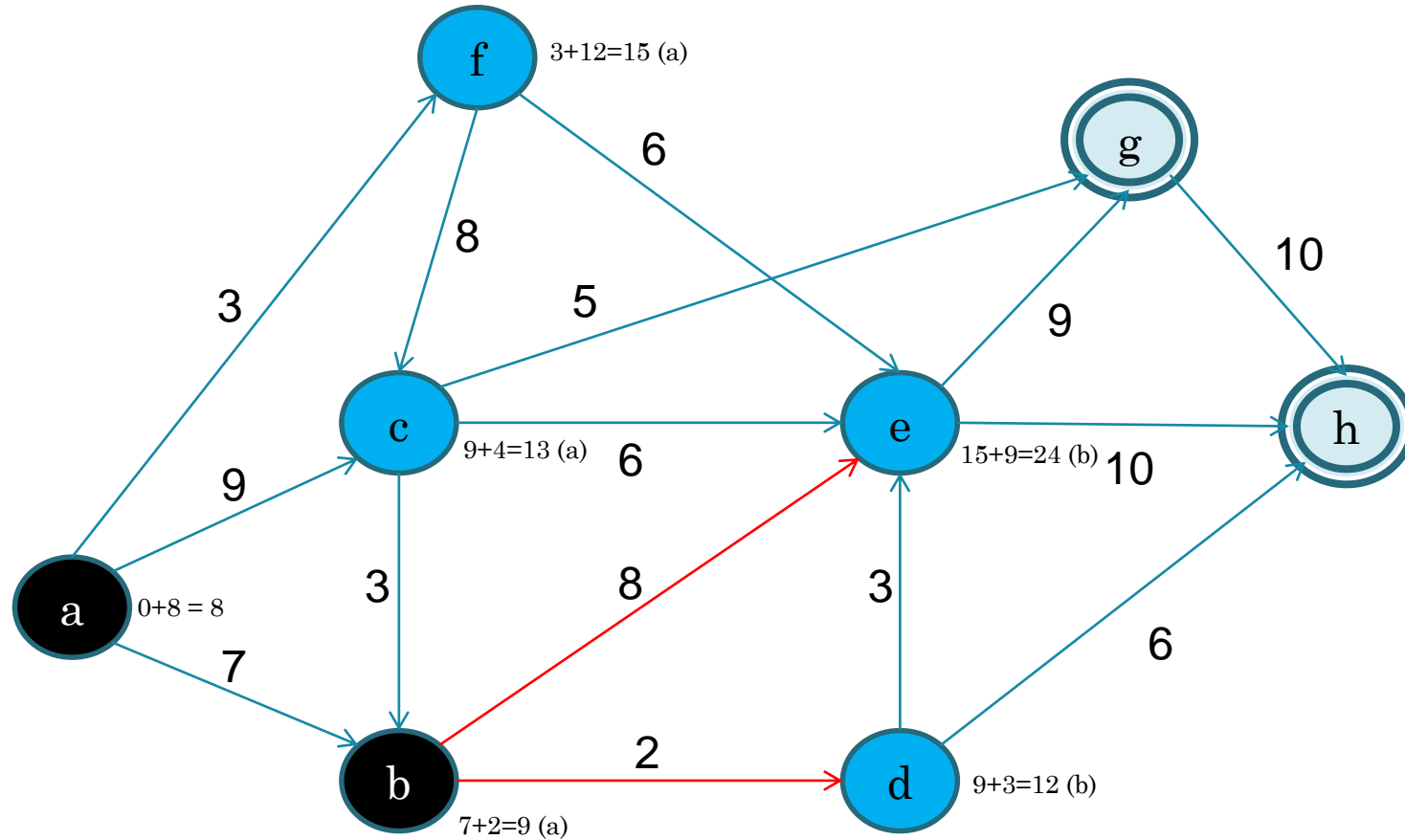
	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – A* (3/8)



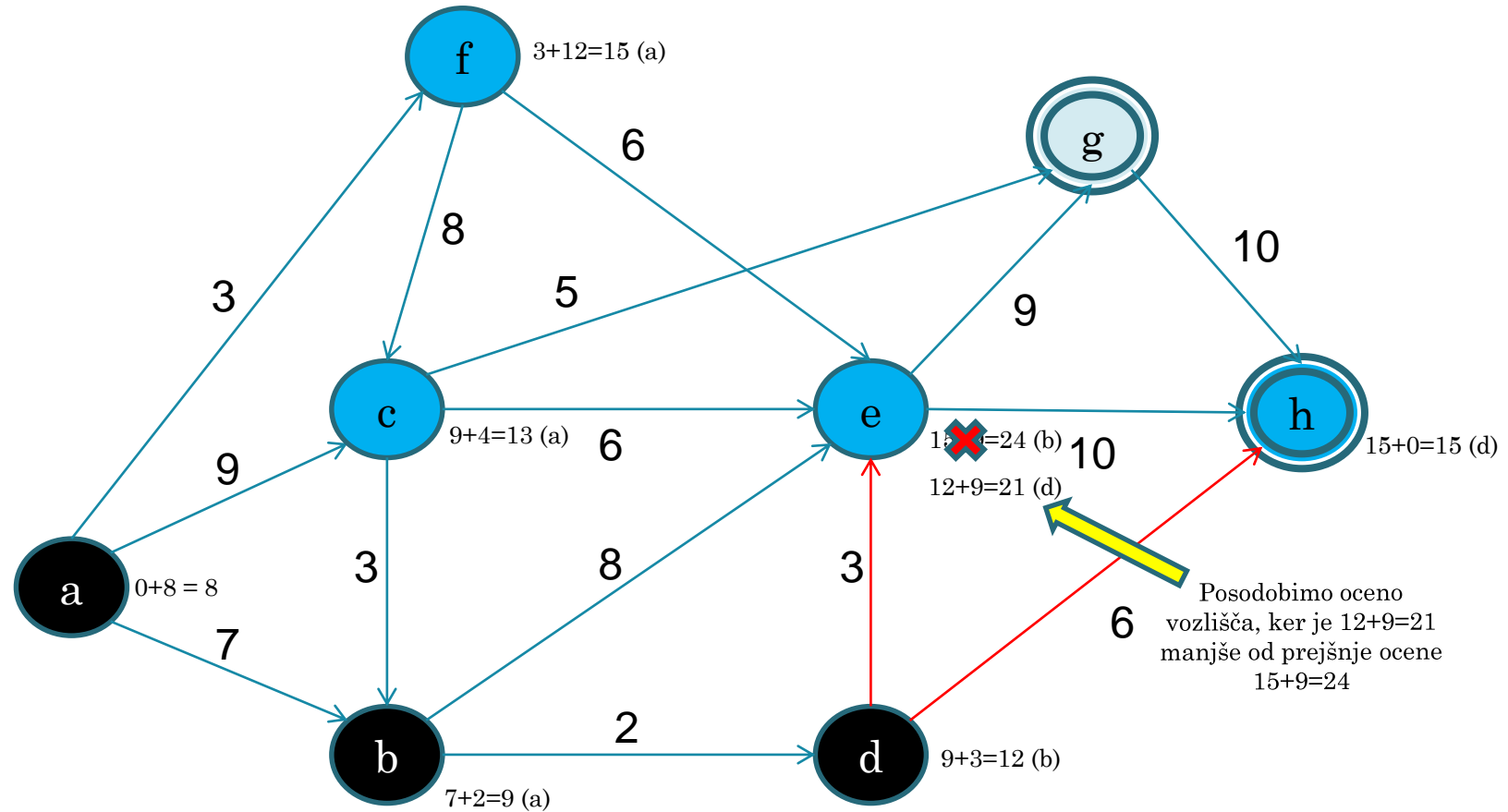
	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – A* (4/8)



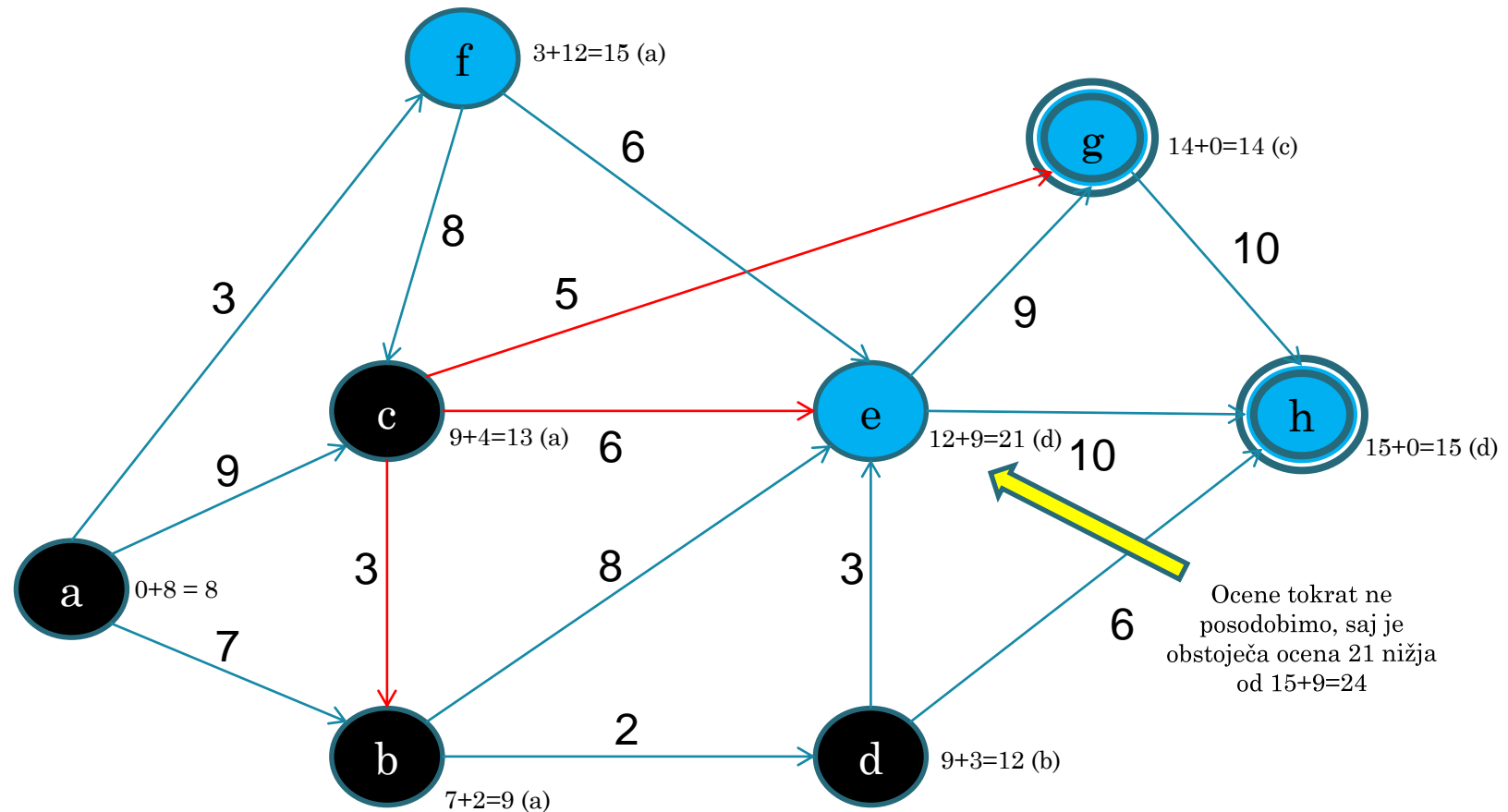
	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – A* (5/8)



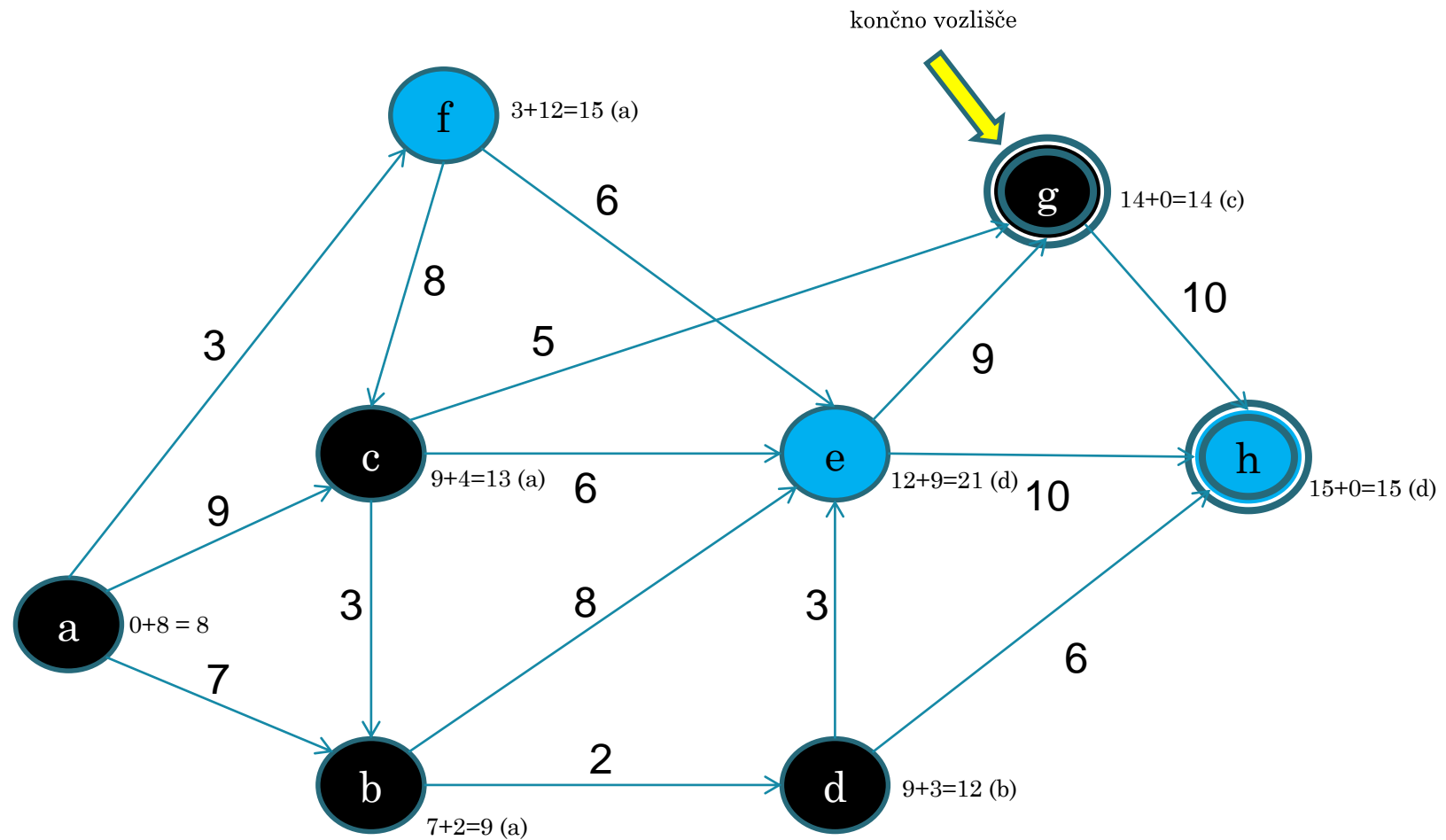
	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – A* (6/8)



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

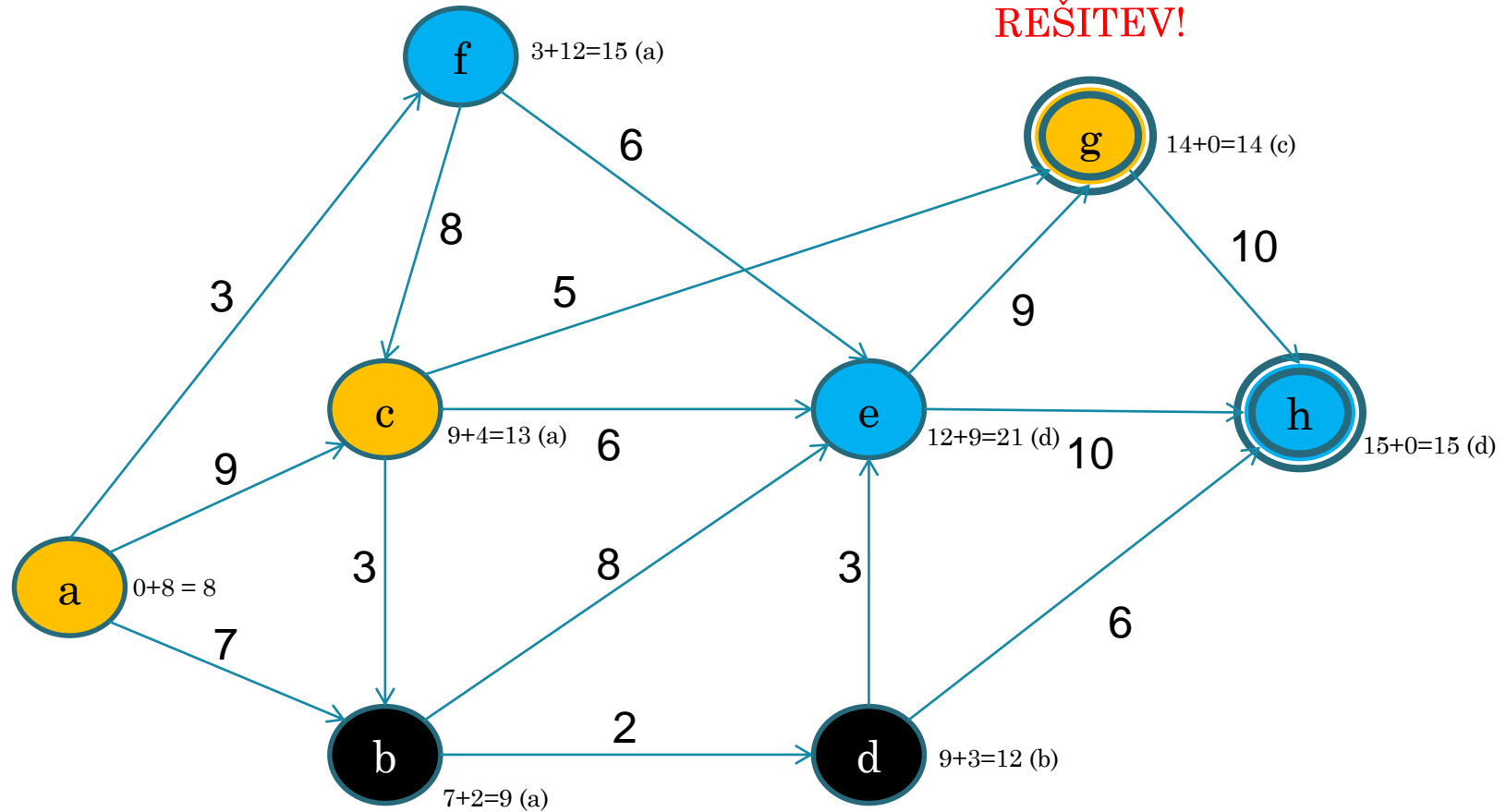
PRIMER – A* (7/8)



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – A* (8/8)

DOBLJENA POT JE OPTIMALNA
REŠITEV!



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

IZBOLJŠAVE ALGORITMA A*

Težava algoritma A* je prevelika poraba pomnilnika.

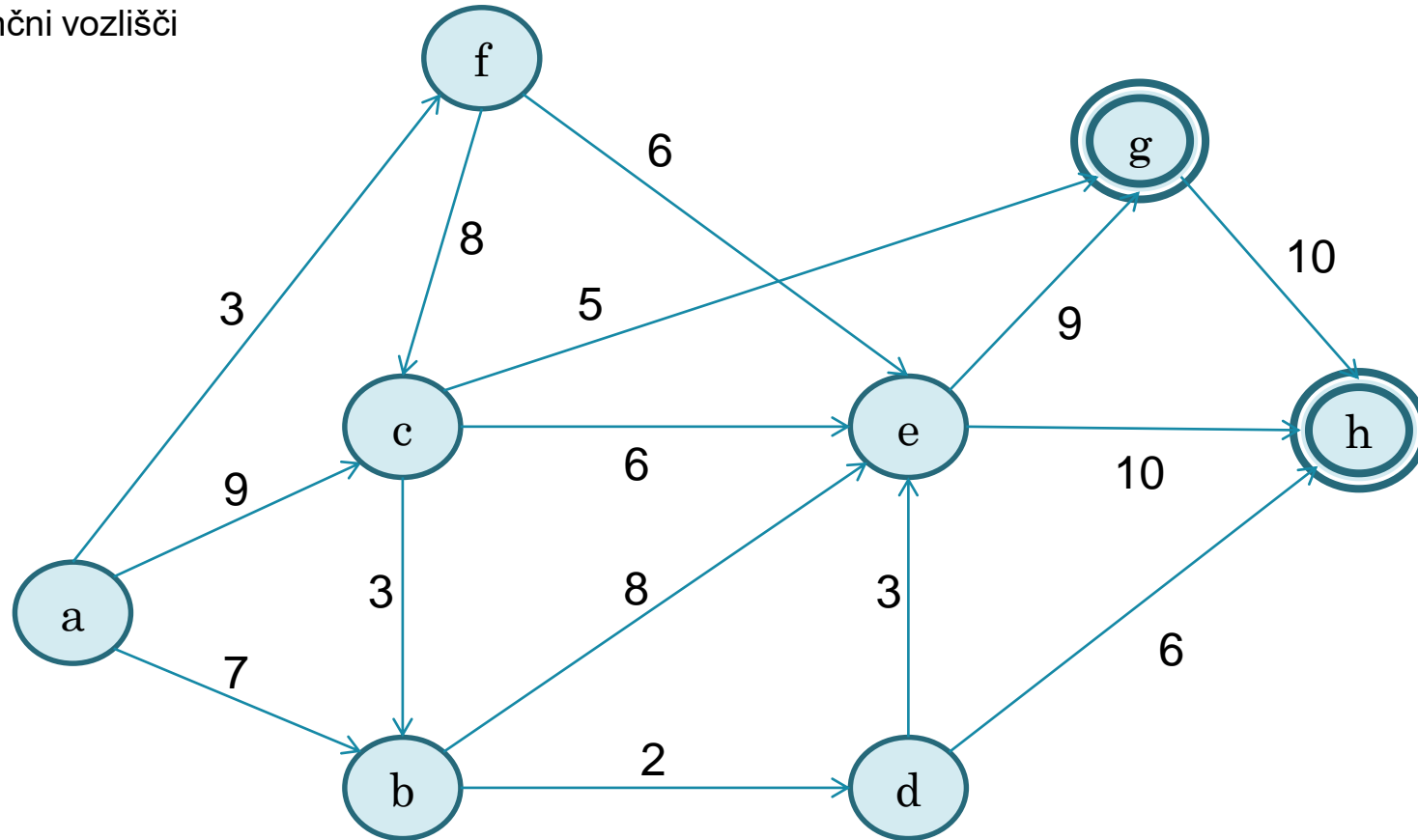
Izboljšave porabijo manj pomnilnika, a še vedno zagotavljajo optimalnost rešitve.

Iterative-deepening A* (IDA*):

- namesto povečevanja globine iskanja, povečuje vrednost hevristične ocene $f(n)$

PRIMER – IDA* (1/36)

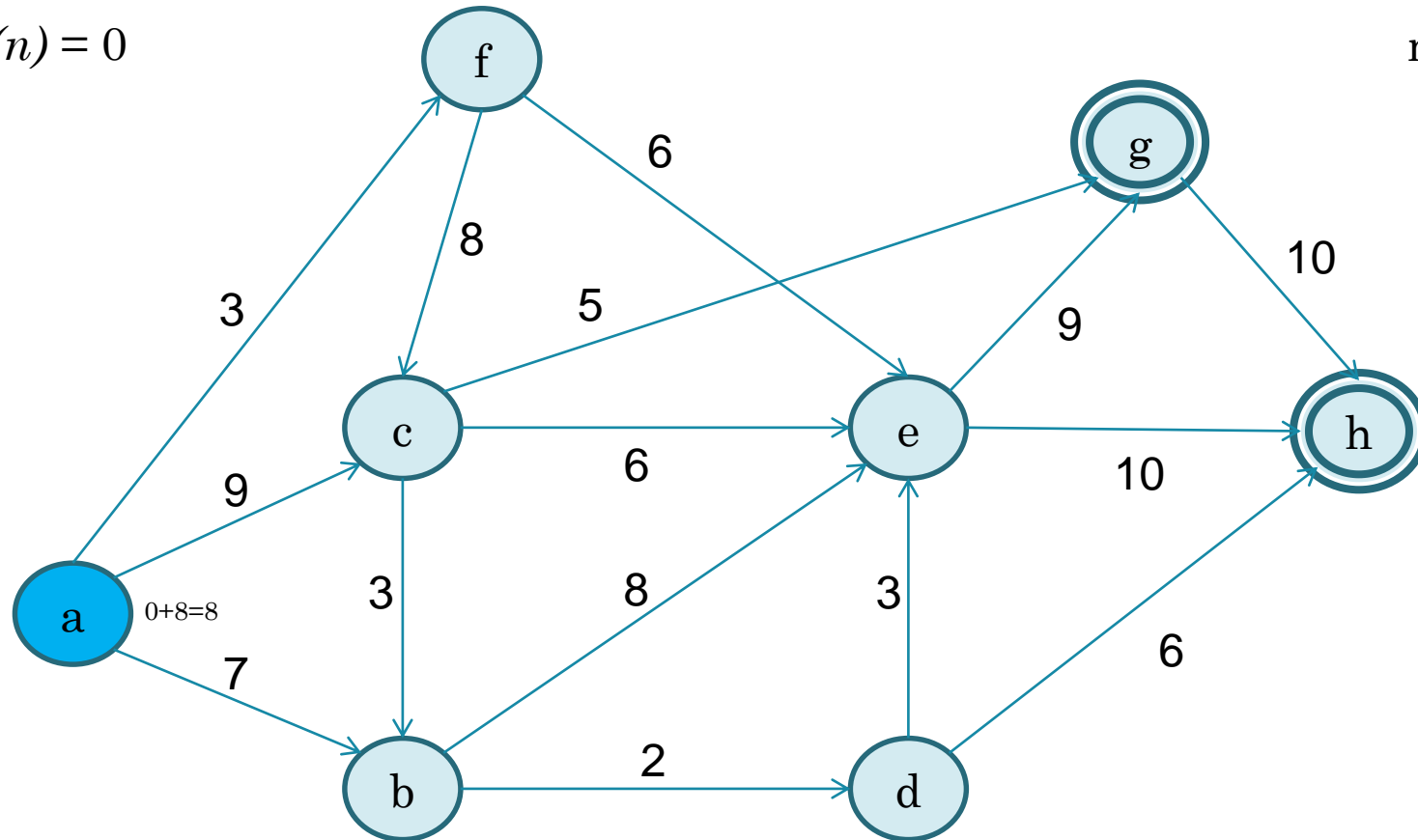
a – začetno vozlišče
g,h – končni vozlišči



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (2/36)

Meja $f(n) = 0$



min. čez mejo = 8

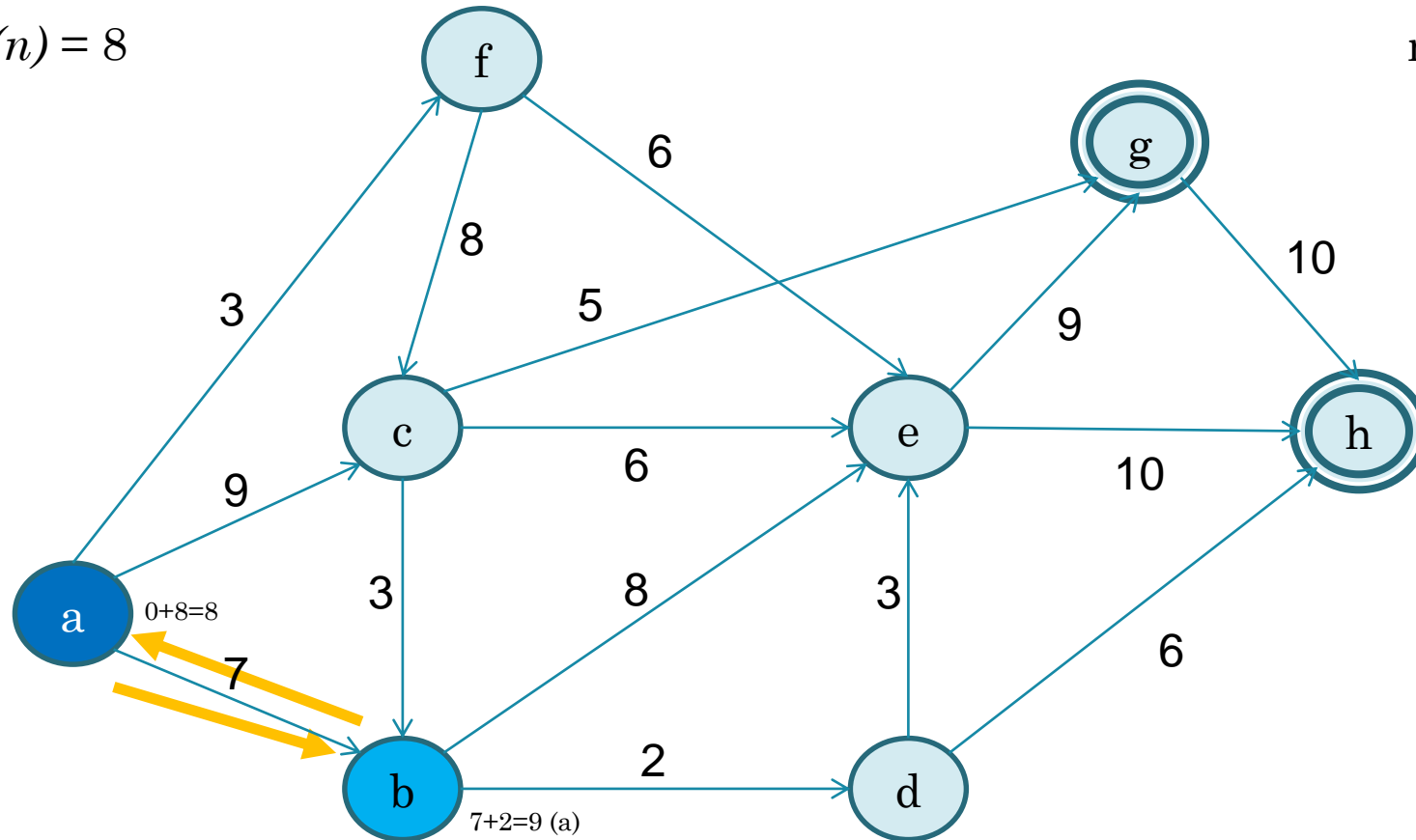
↑
nova meja

	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (3/36)

Meja $f(n) = 8$

min. čez mejo = 9

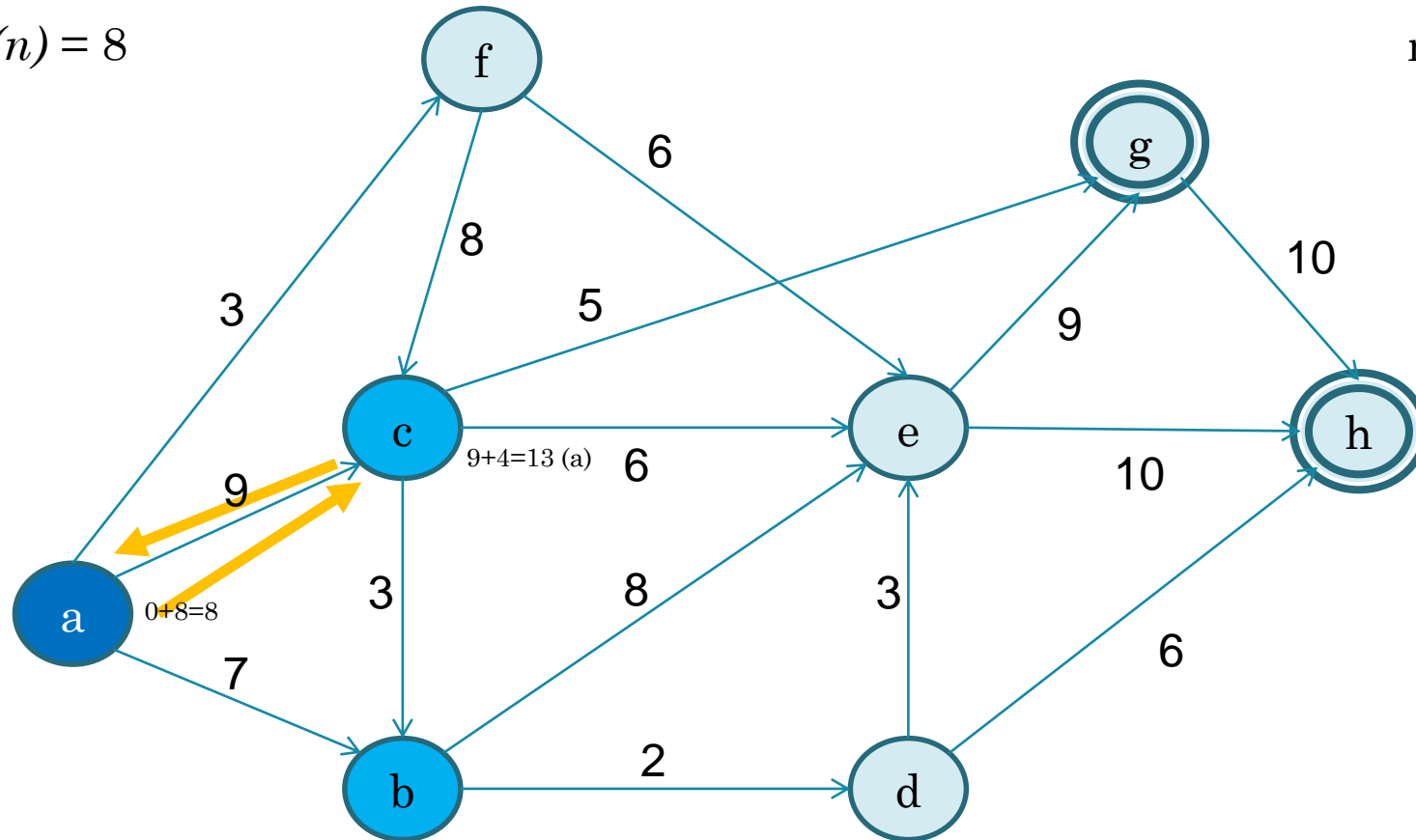


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (4/36)

Meja $f(n) = 8$

min. čez mejo = 9

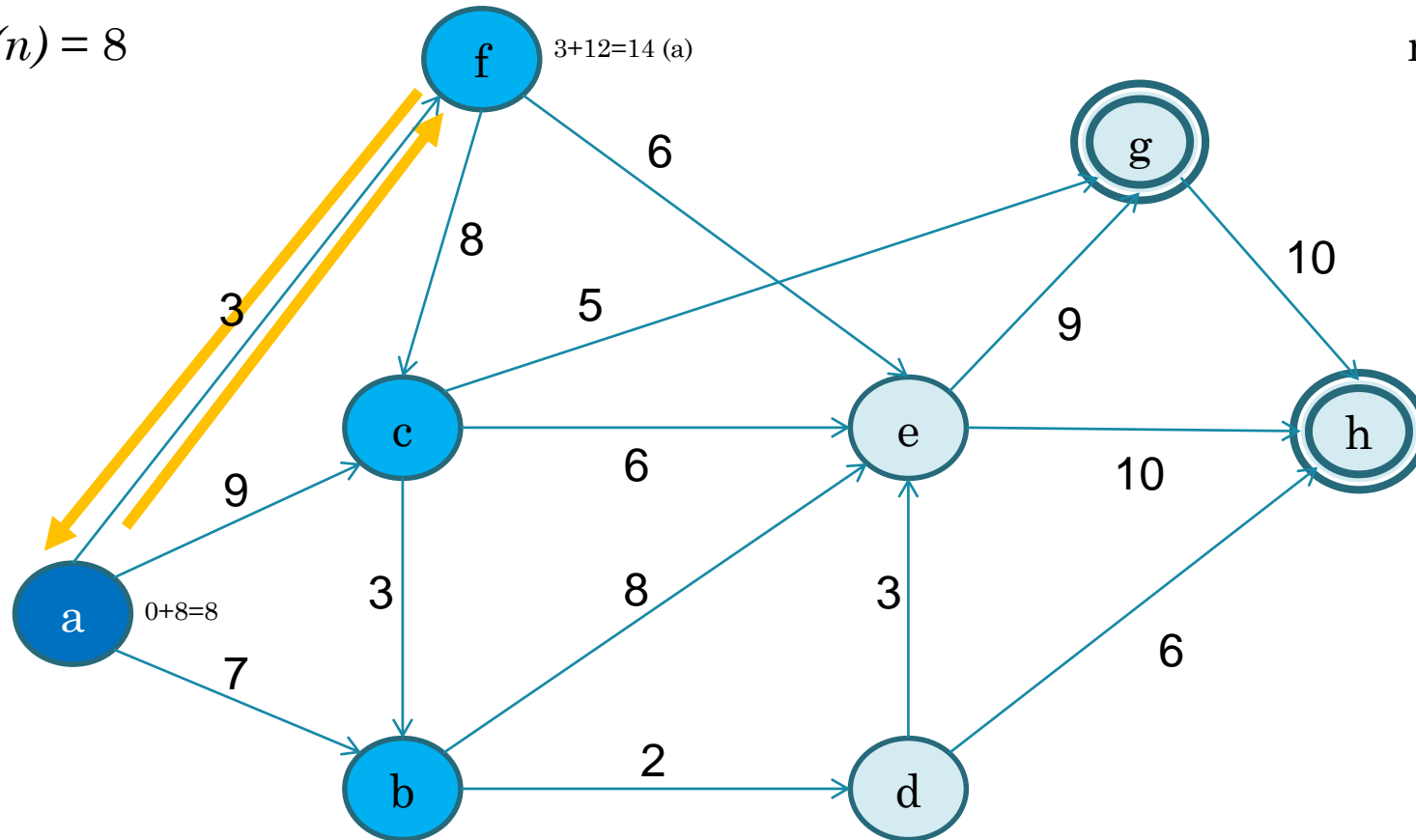


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (5/36)

Meja $f(n) = 8$

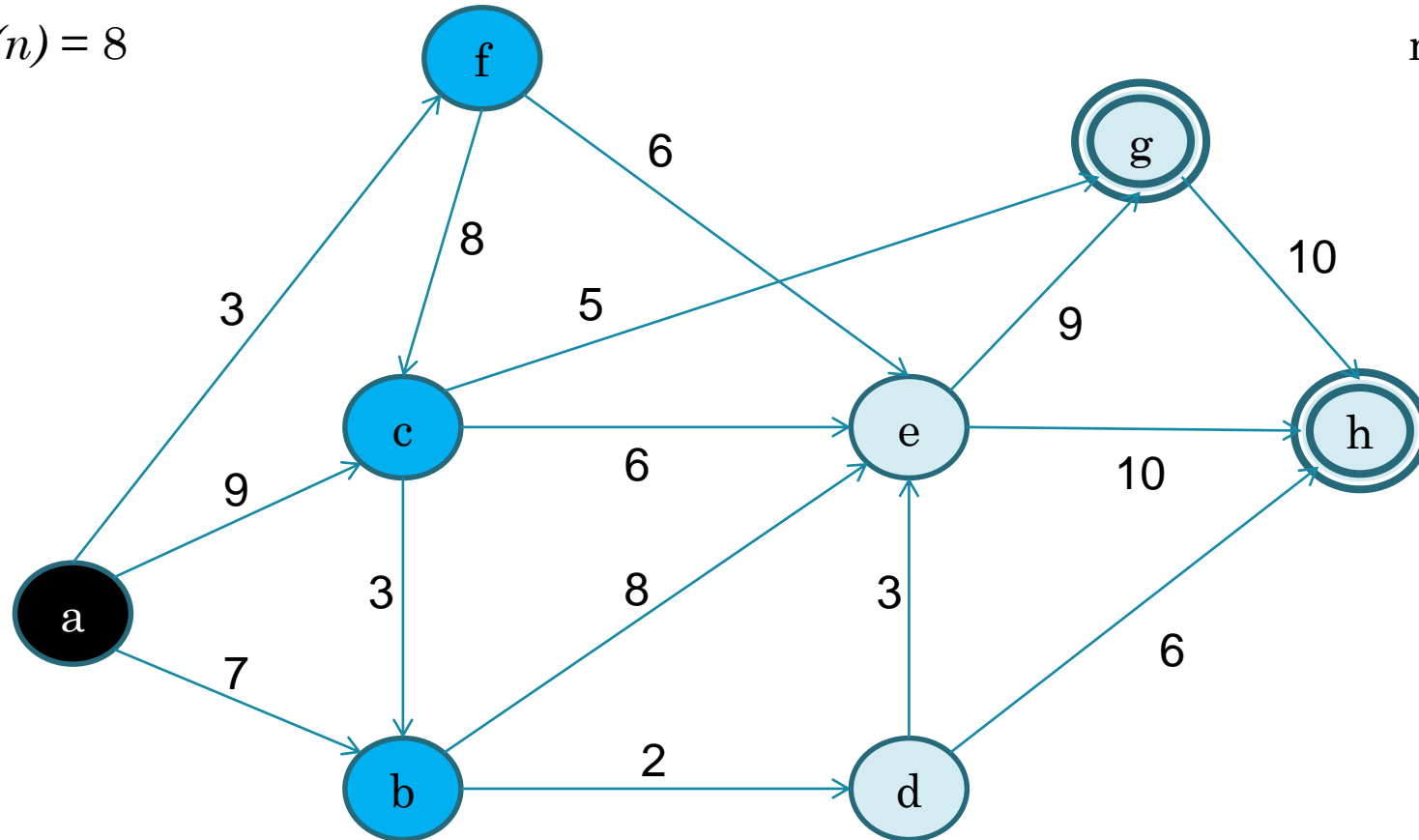
min. čez mejo = 9



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (6/36)

Meja $f(n) = 8$



min. čez mejo = 9

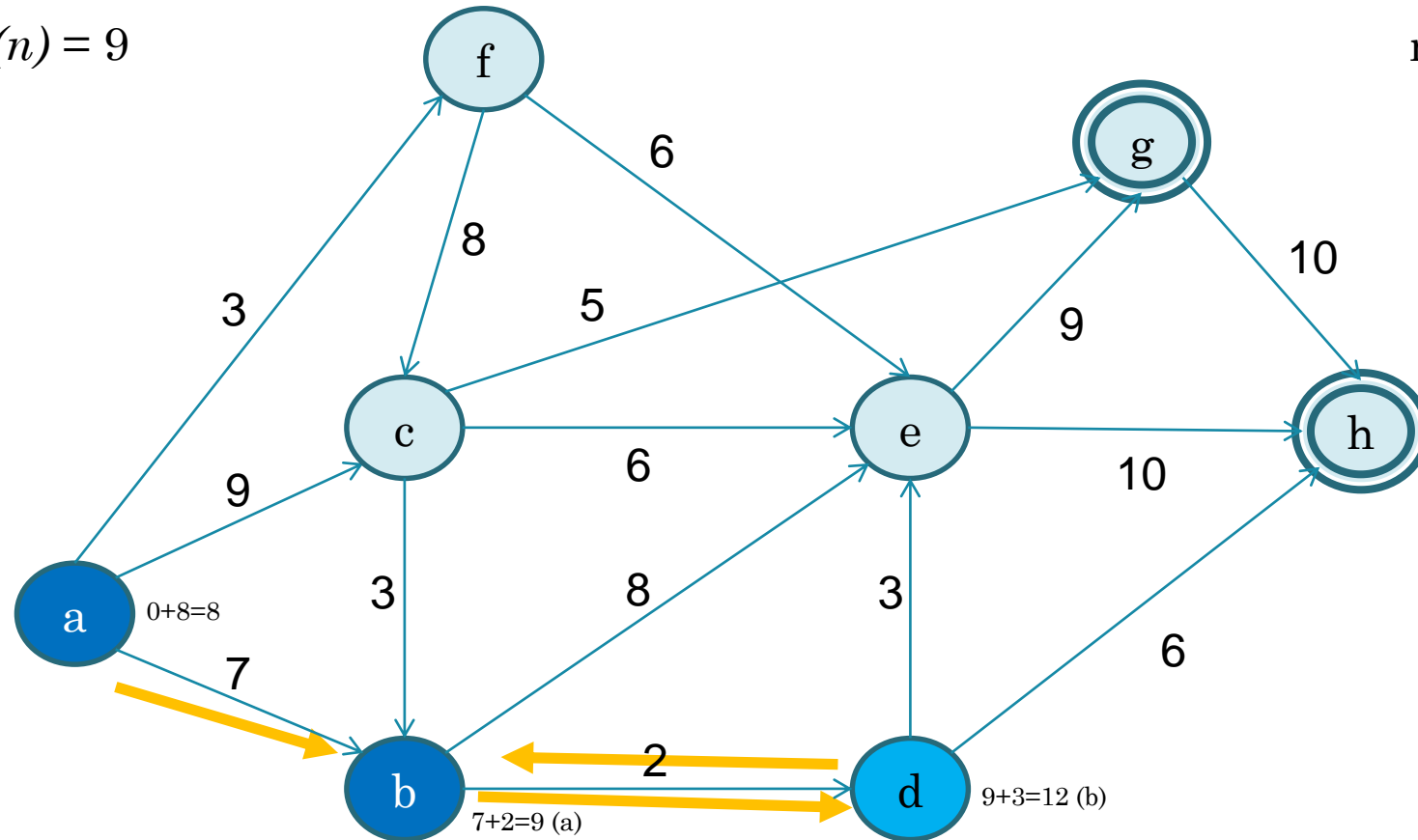
↑
nova meja

	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (7/36)

Meja $f(n) = 9$

min. čez mejo = 12

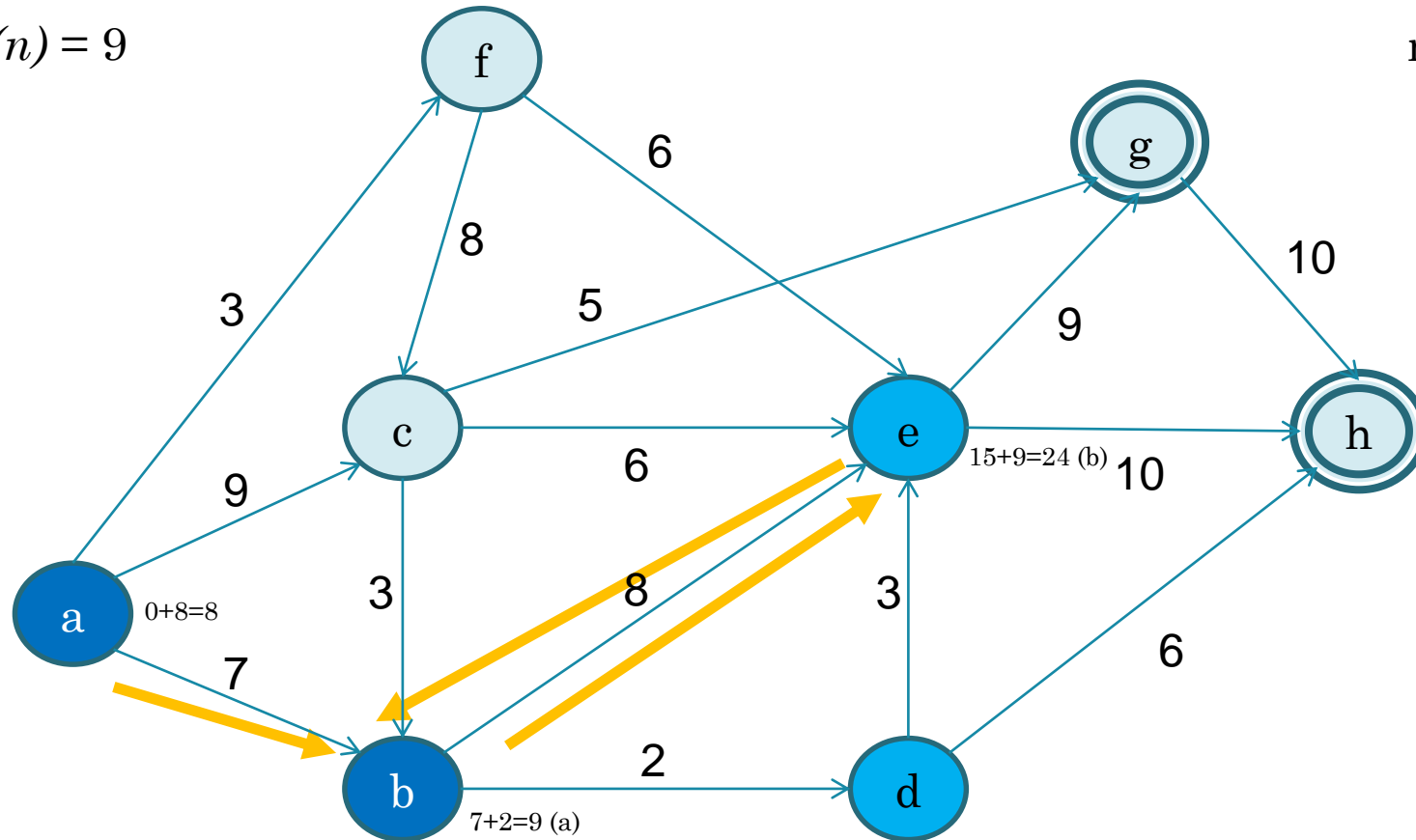


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (8/36)

Meja $f(n) = 9$

min. čez mejo = 12

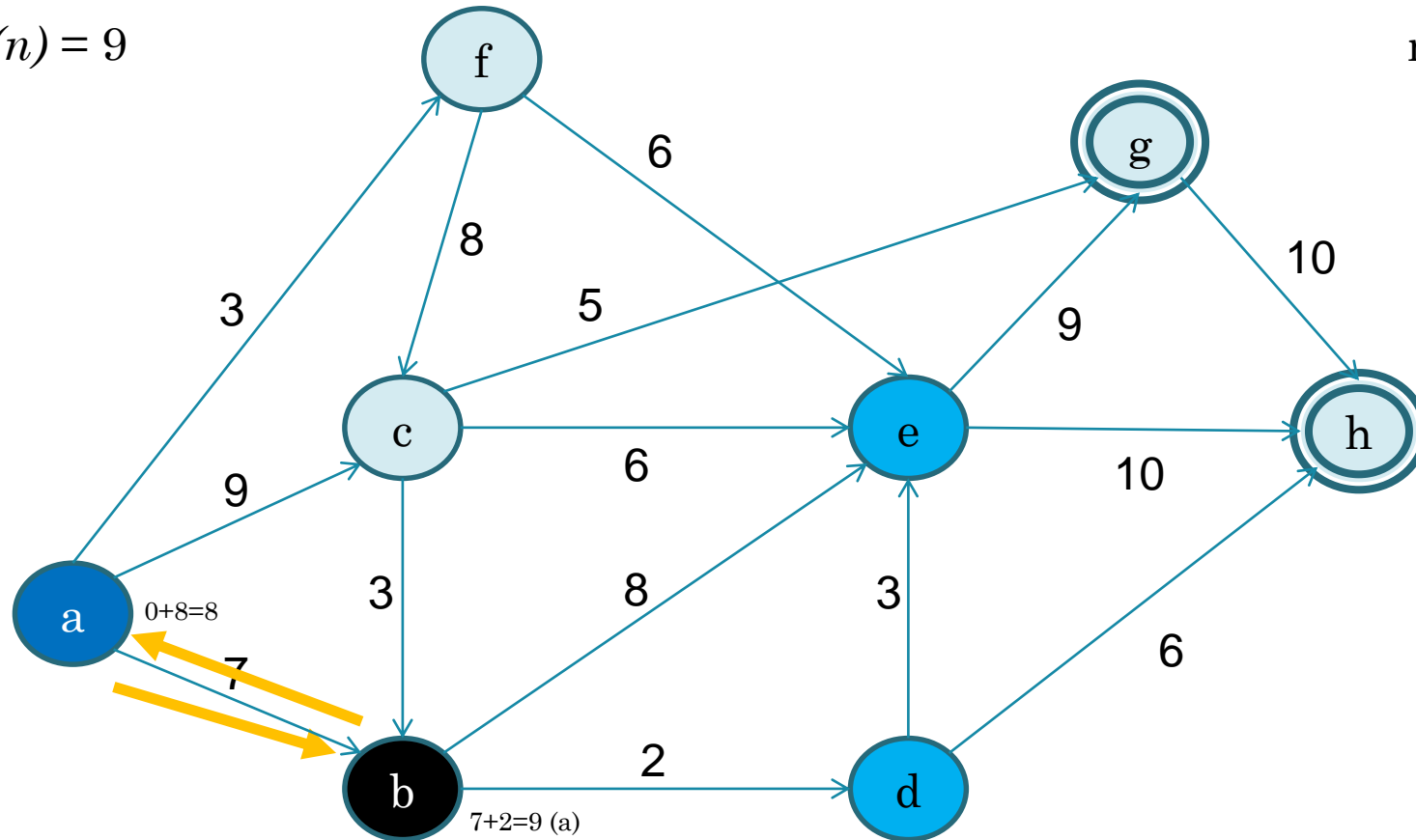


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (9/36)

Meja $f(n) = 9$

min. čez mejo = 12

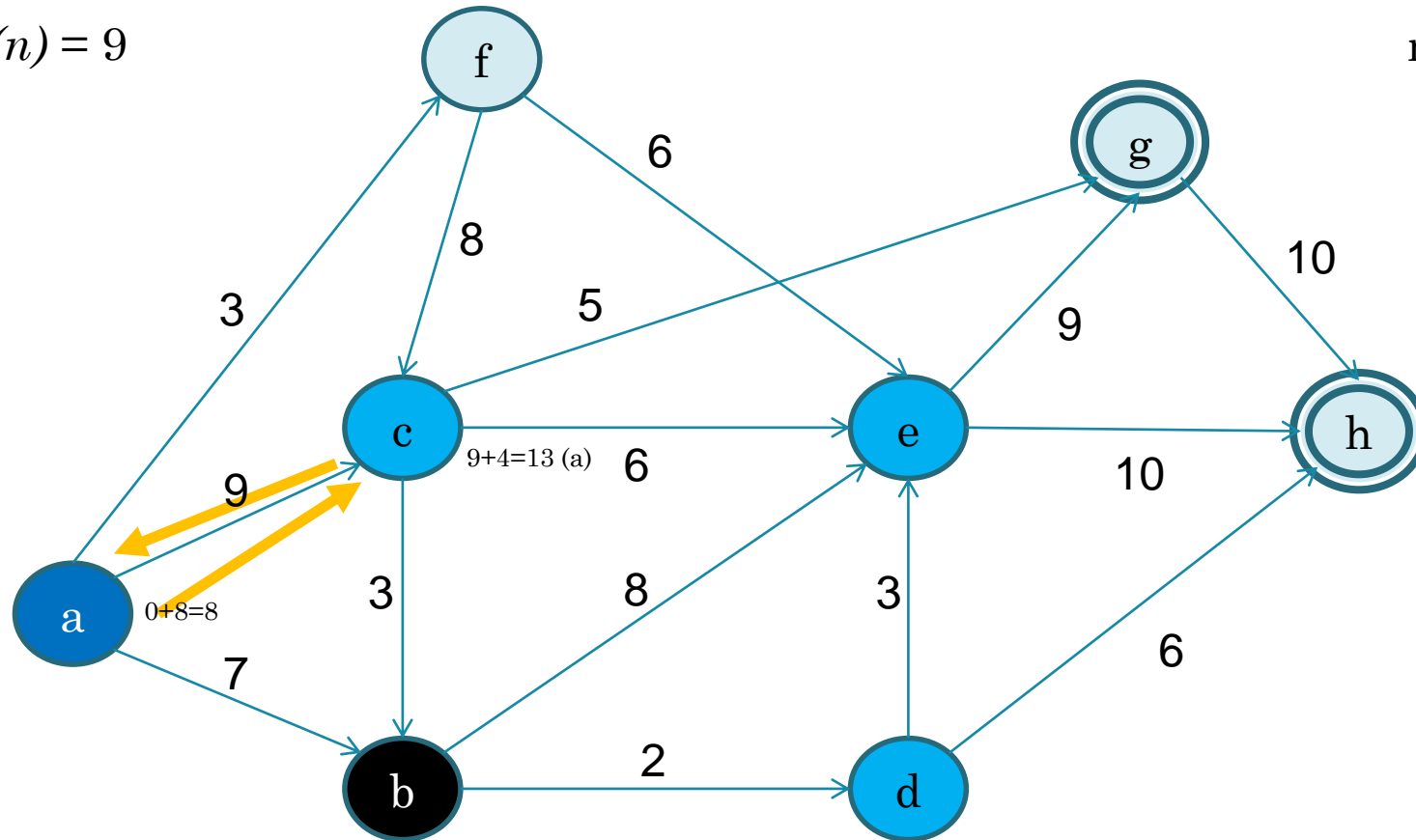


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (10/36)

Meja $f(n) = 9$

min. čez mejo = 12

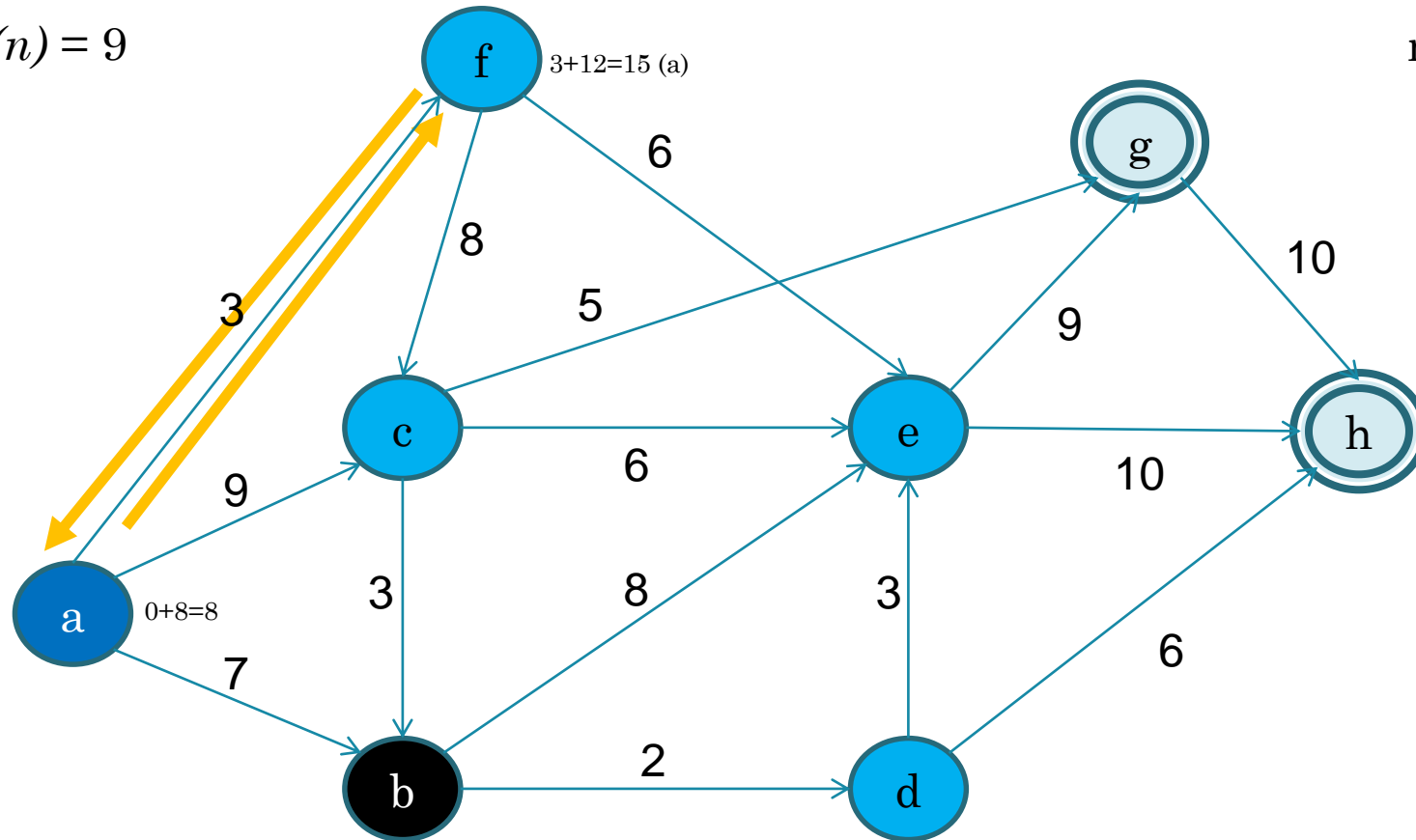


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (11/36)

Meja $f(n) = 9$

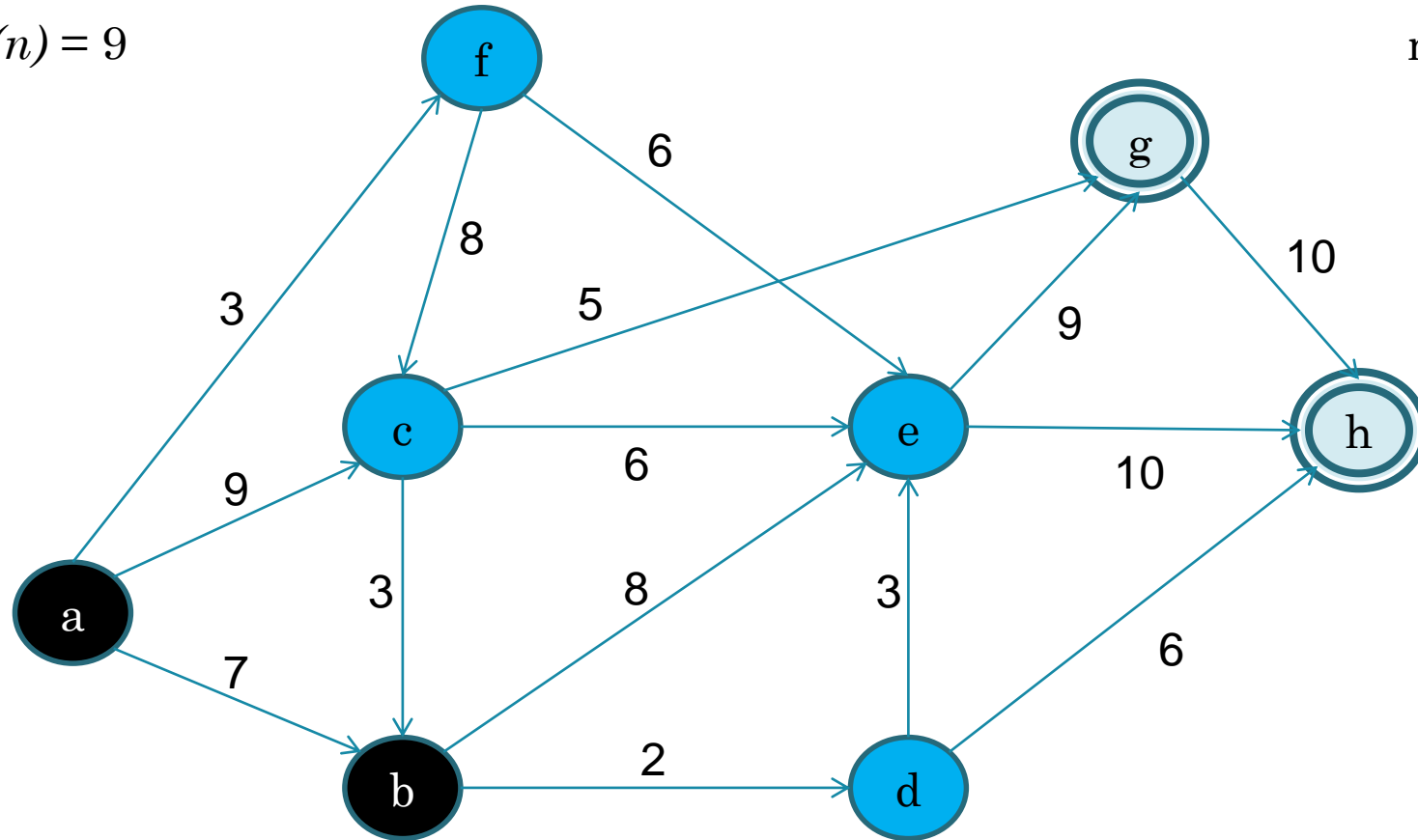
min. čez mejo = 12



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (12/36)

Meja $f(n) = 9$



min. čez mejo = 12



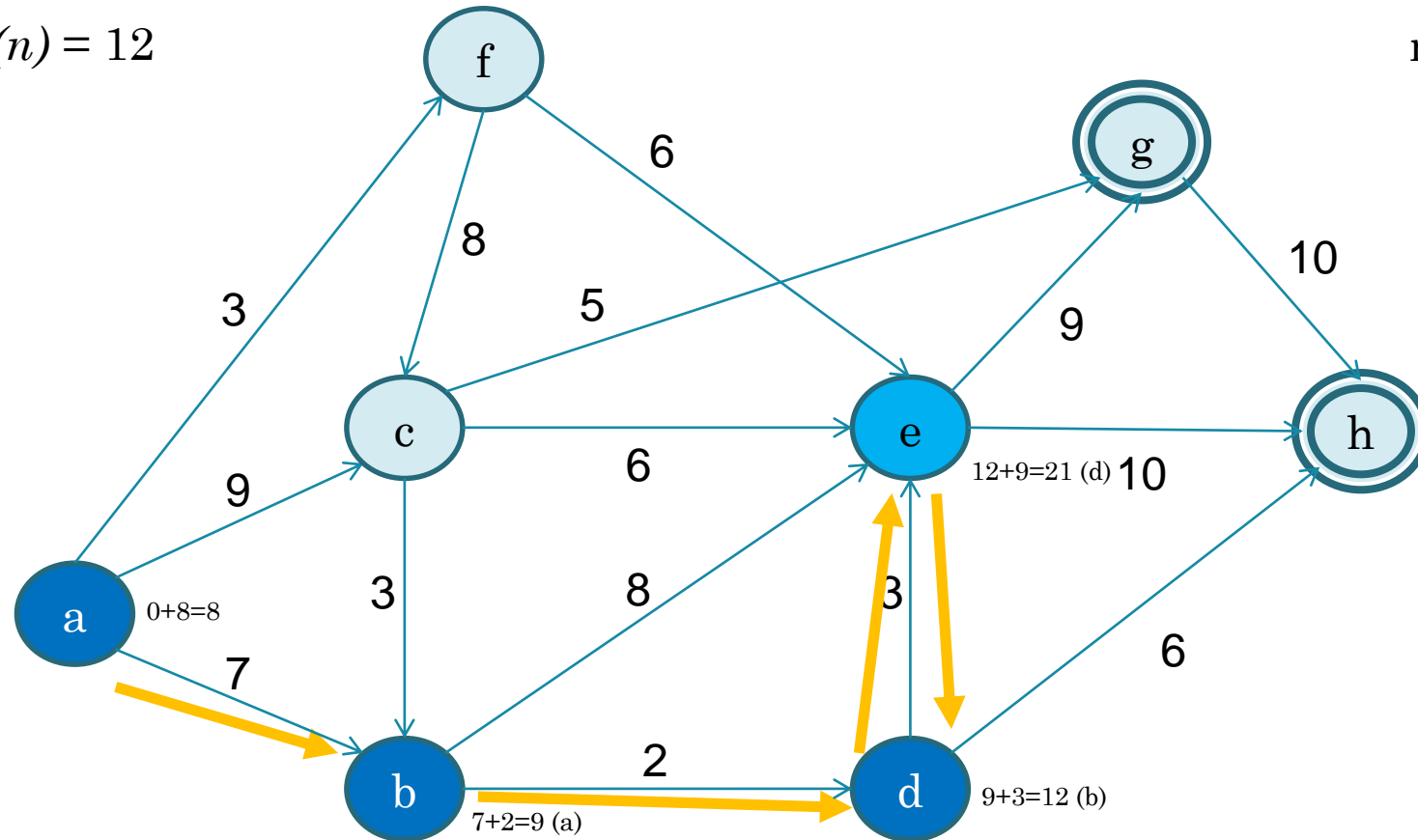
nova meja

	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (13/36)

Meja $f(n) = 12$

min. čez mejo = 21

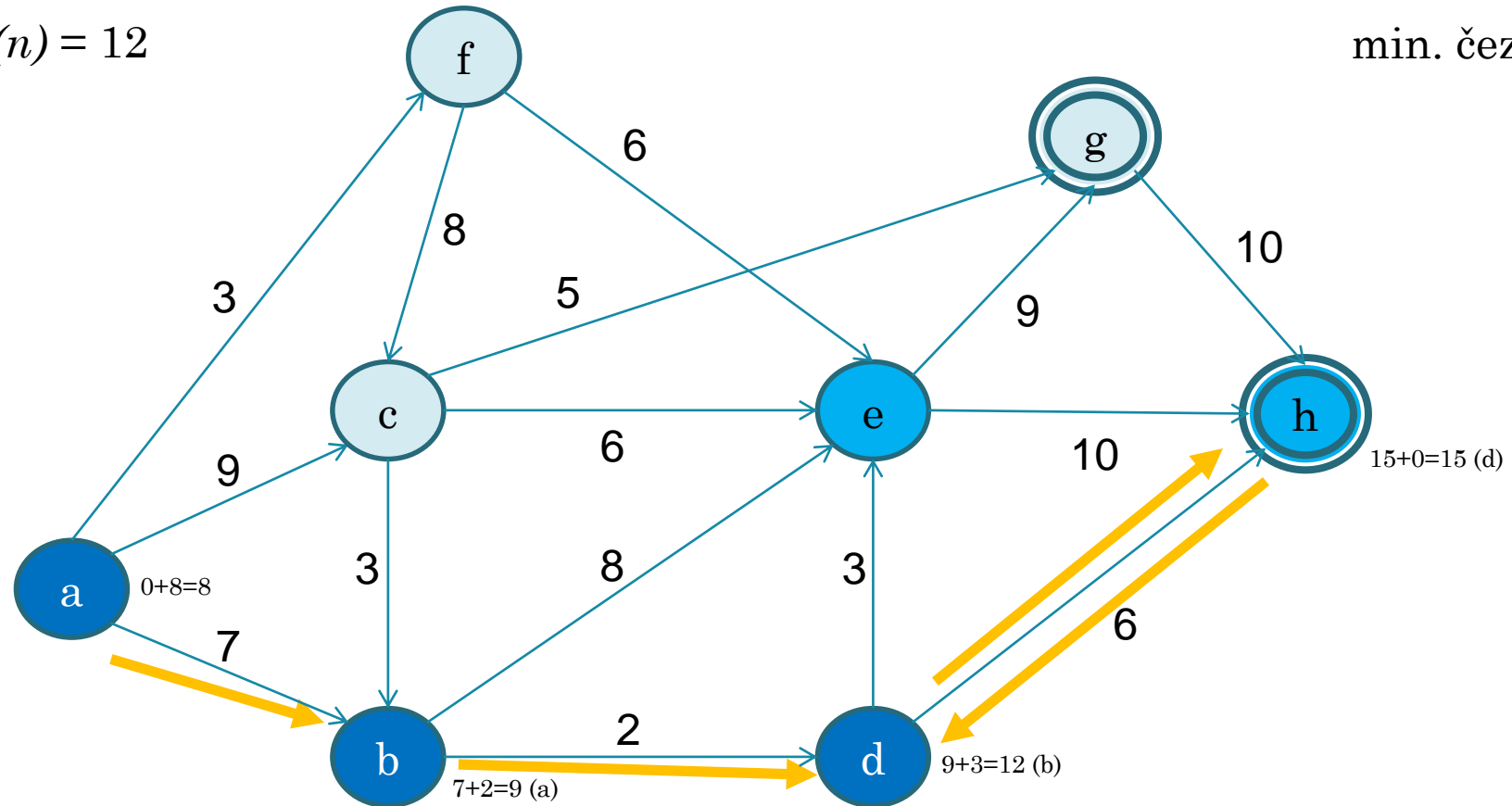


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (14/36)

Meja $f(n) = 12$

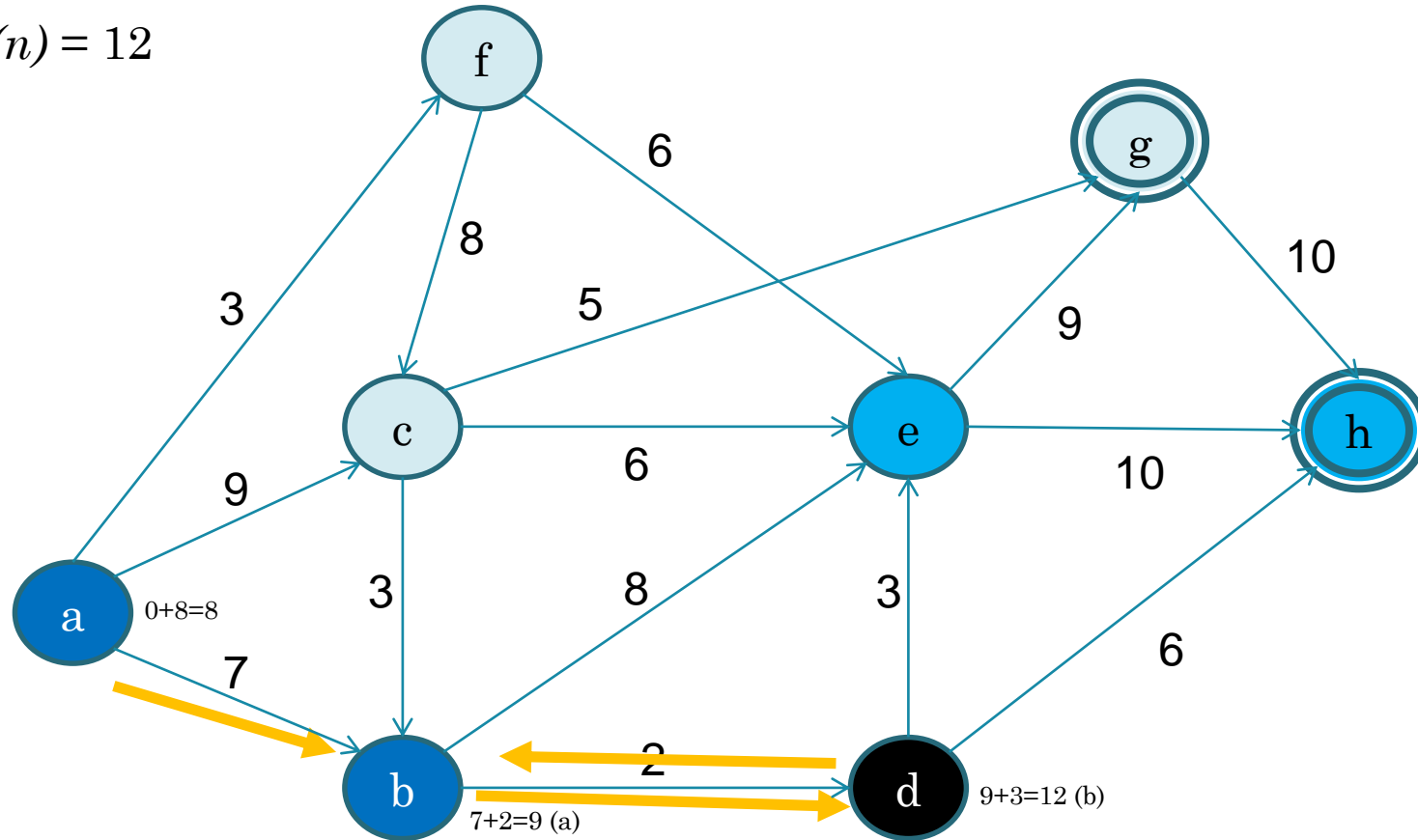
min. čez mejo = 15



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (15/36)

Meja $f(n) = 12$

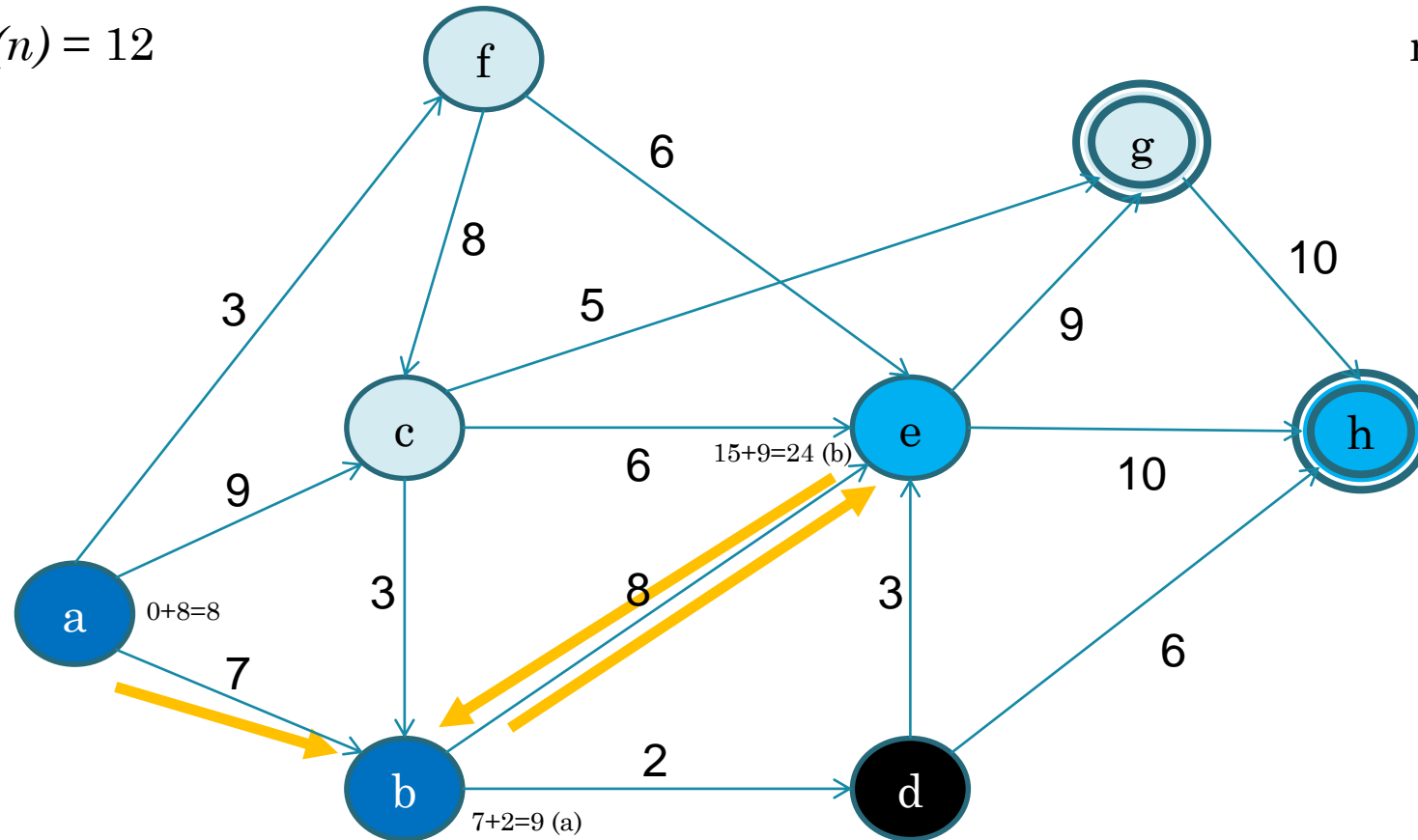


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (16/36)

Meja $f(n) = 12$

min. čez mejo = 15

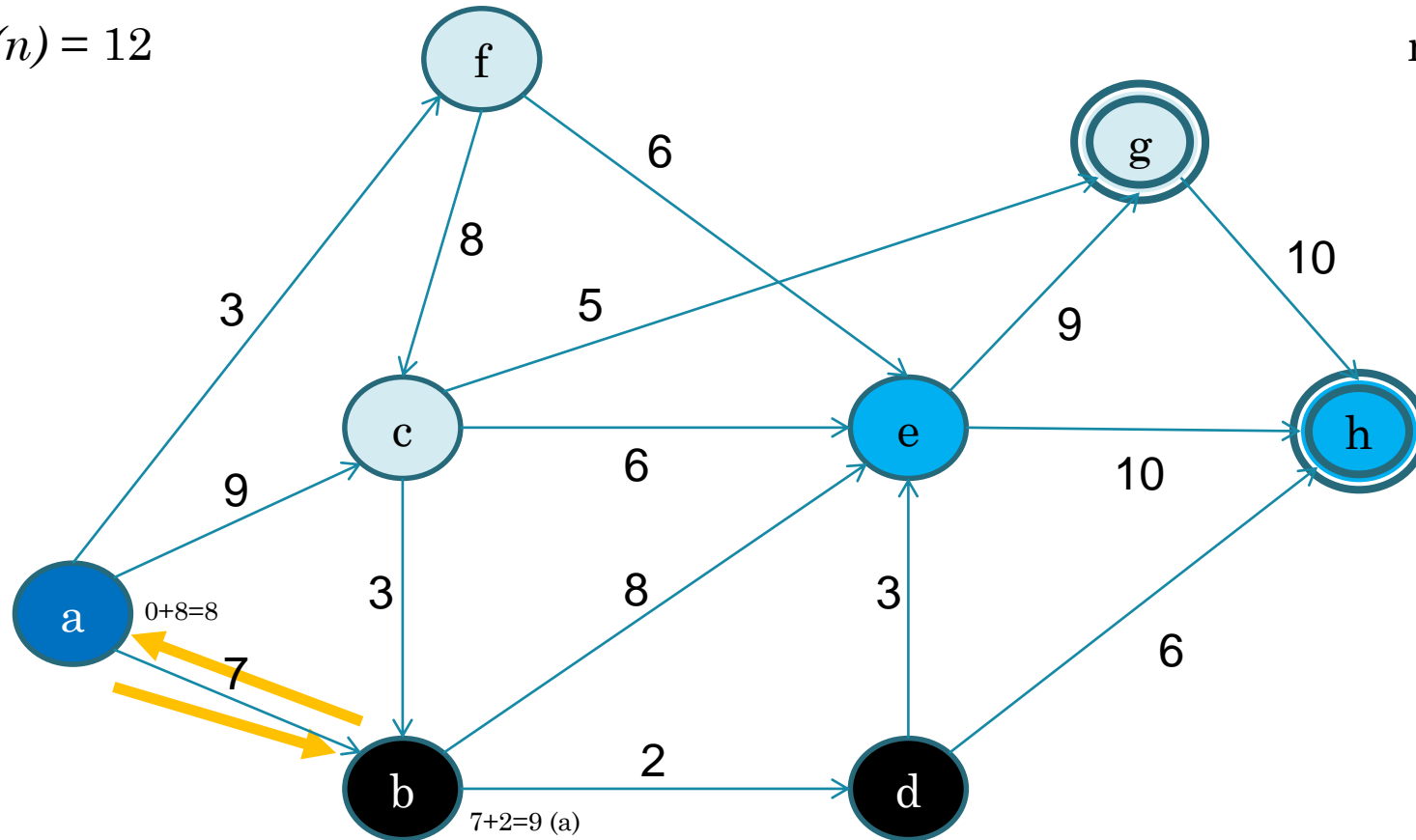


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (17/36)

Meja $f(n) = 12$

min. čez mejo = 15

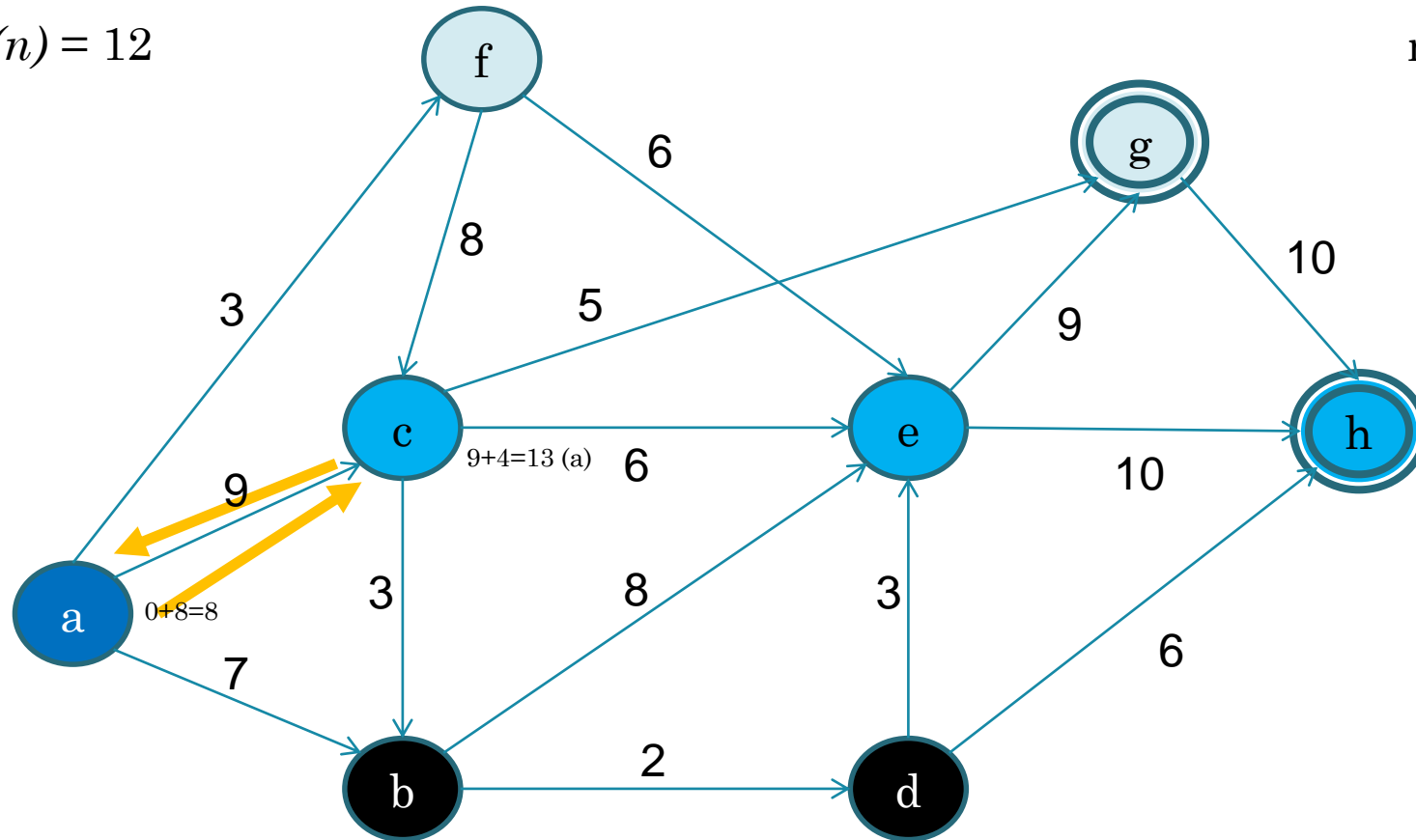


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (18/36)

Meja $f(n) = 12$

min. čez mejo = 13

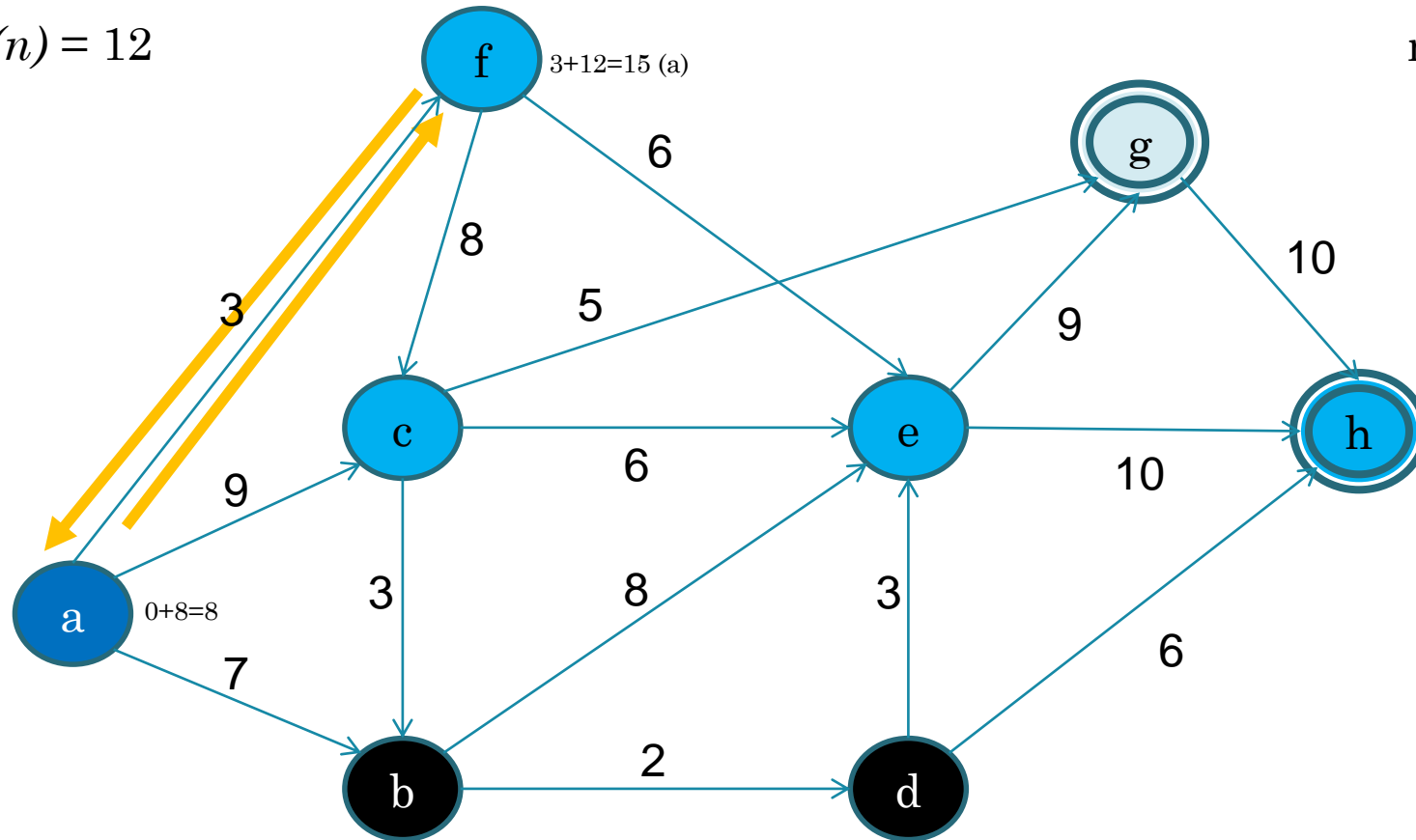


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (19/36)

Meja $f(n) = 12$

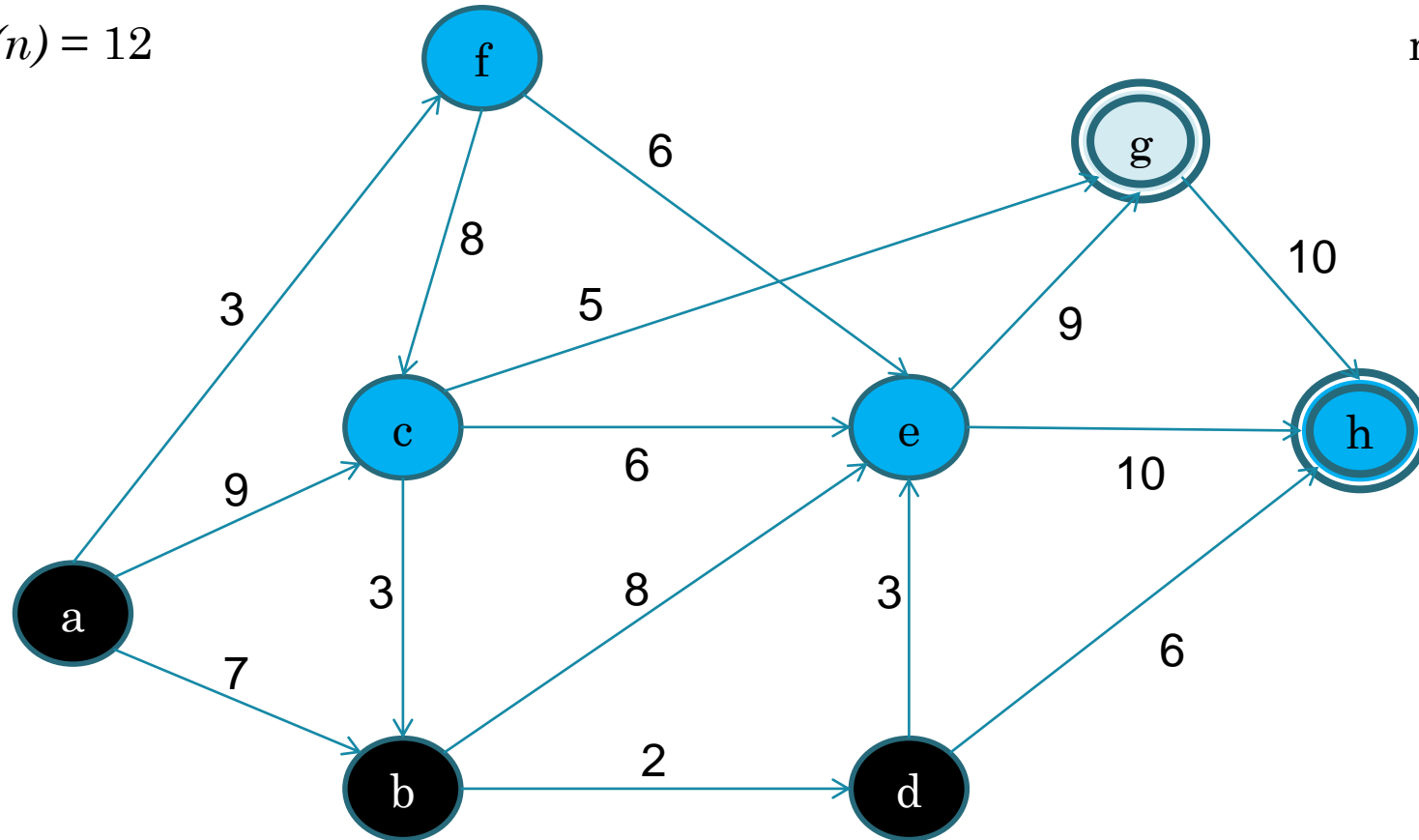
min. čez mejo = 13



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (20/36)

Meja $f(n) = 12$



min. čez mejo = 13

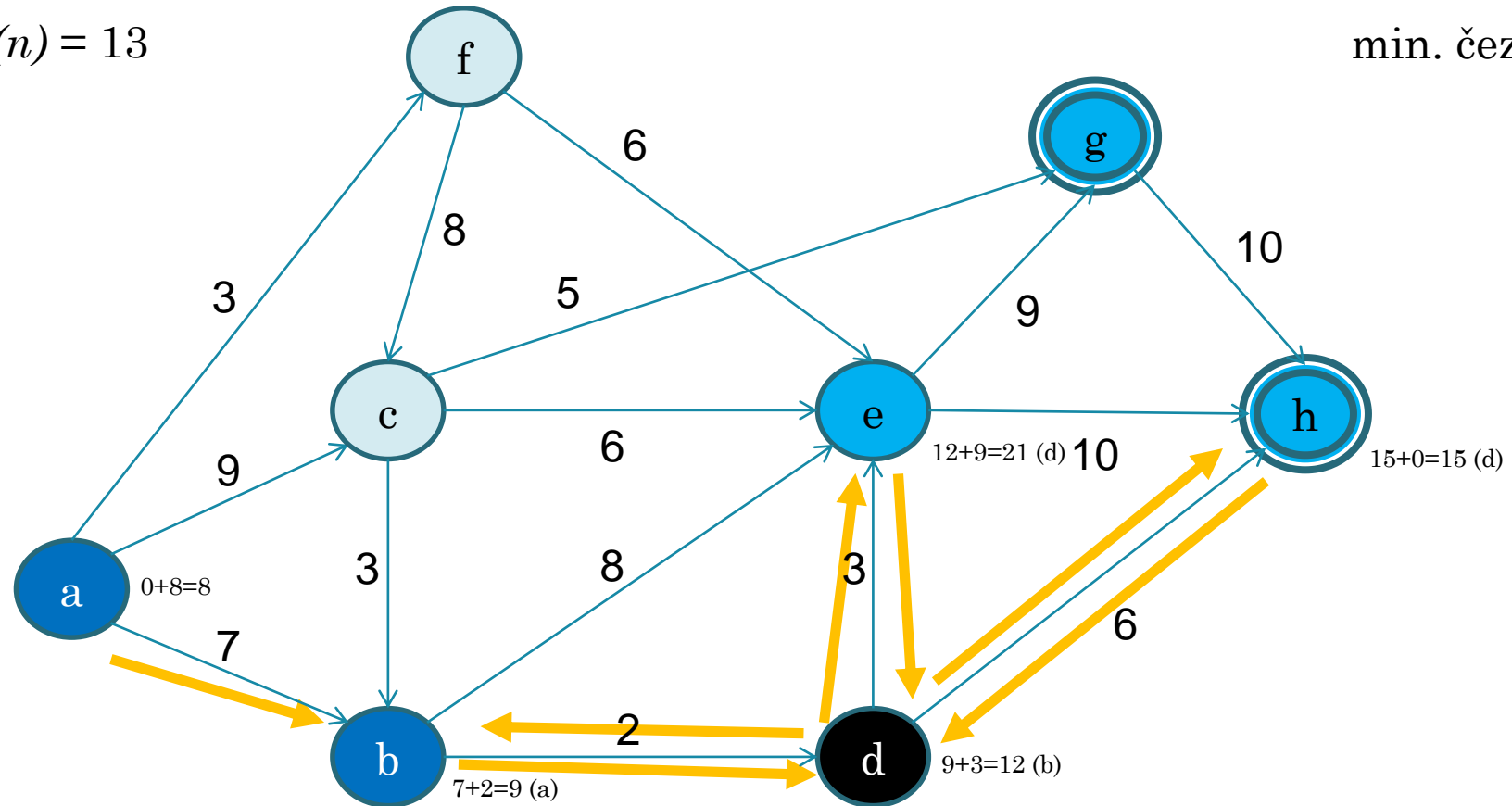
↑
nova meja

	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (21/36)

Meja $f(n) = 13$

min. čez mejo = 15

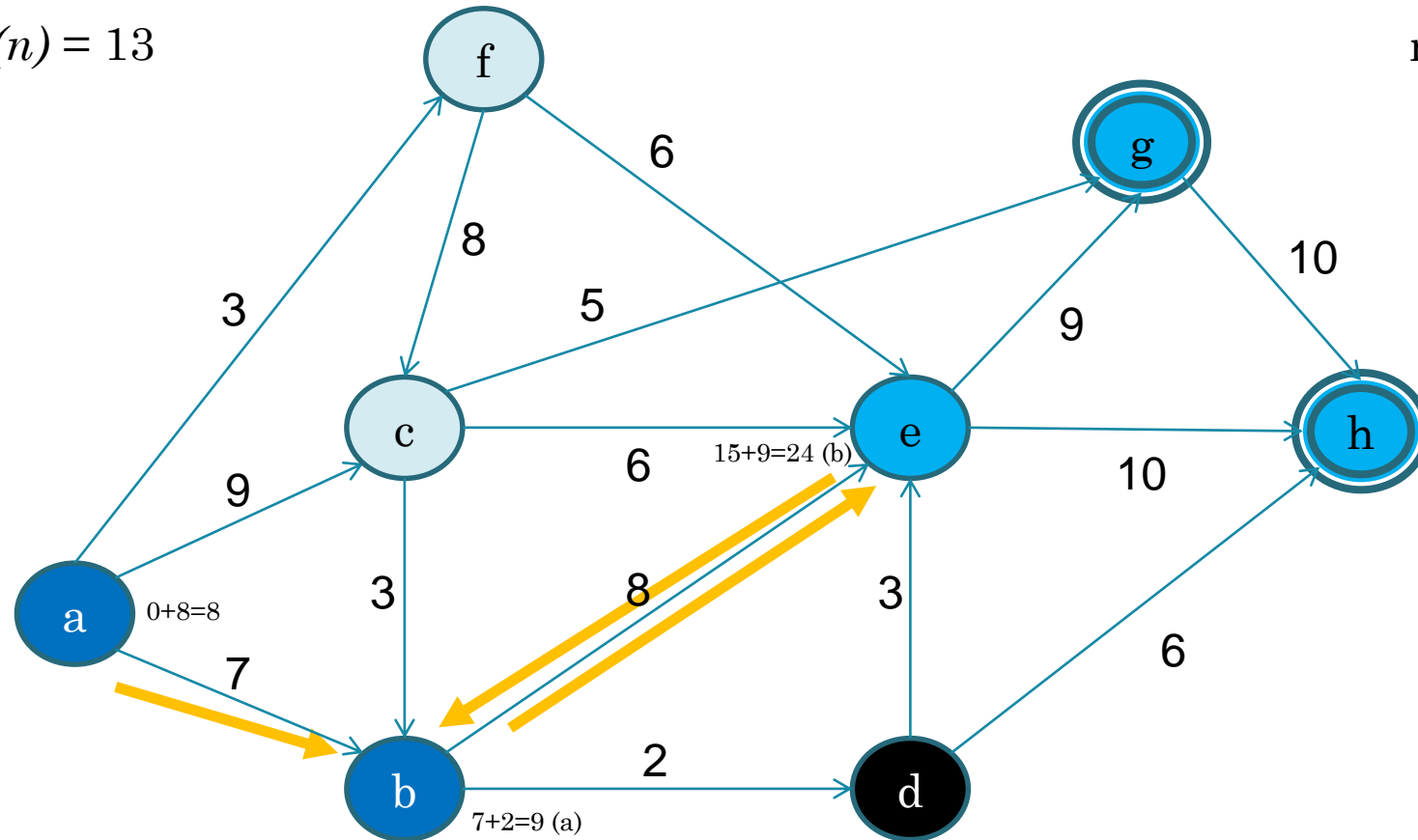


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (22/36)

Meja $f(n) = 13$

min. čez mejo = 15

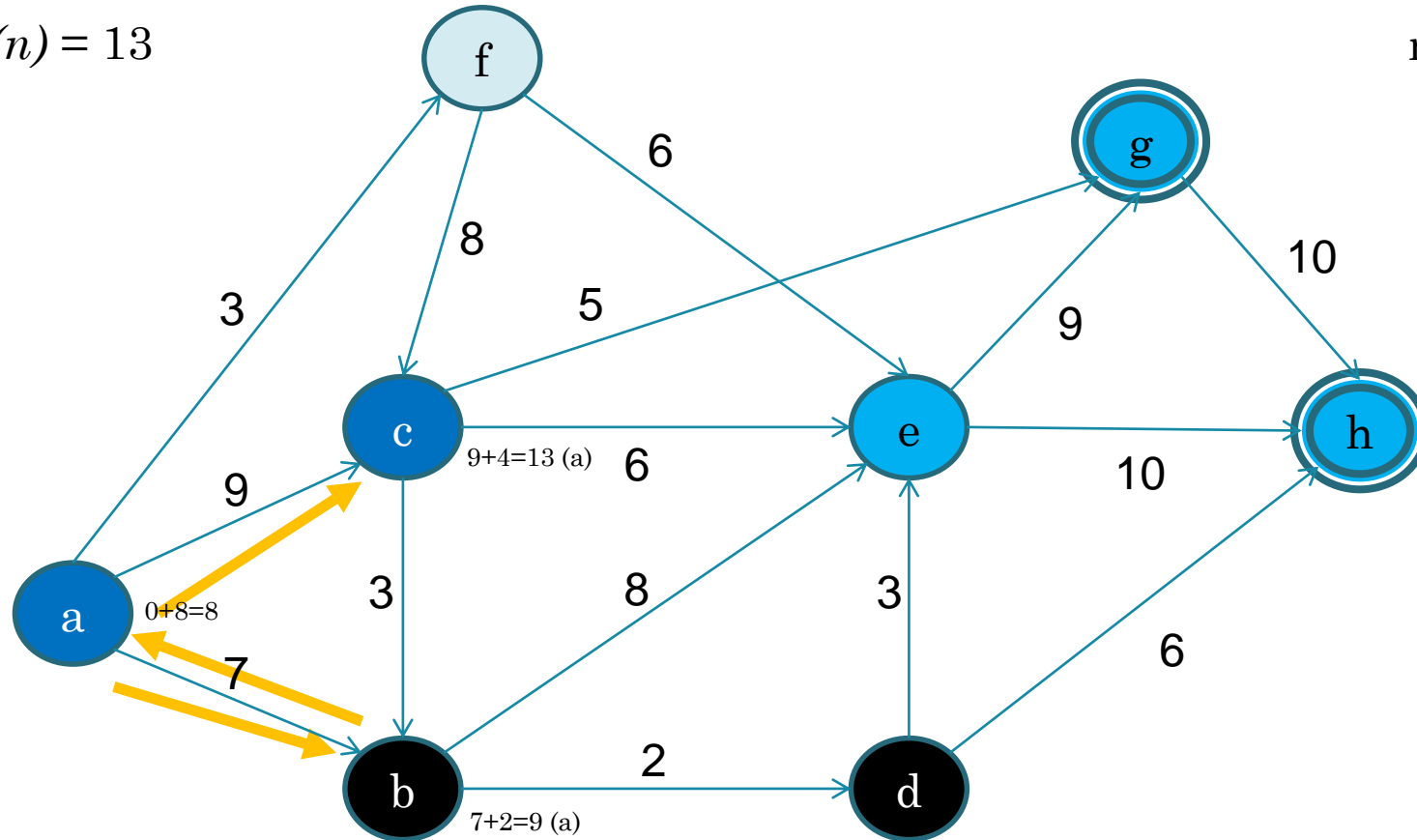


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (23/36)

Meja $f(n) = 13$

min. čez mejo = 15

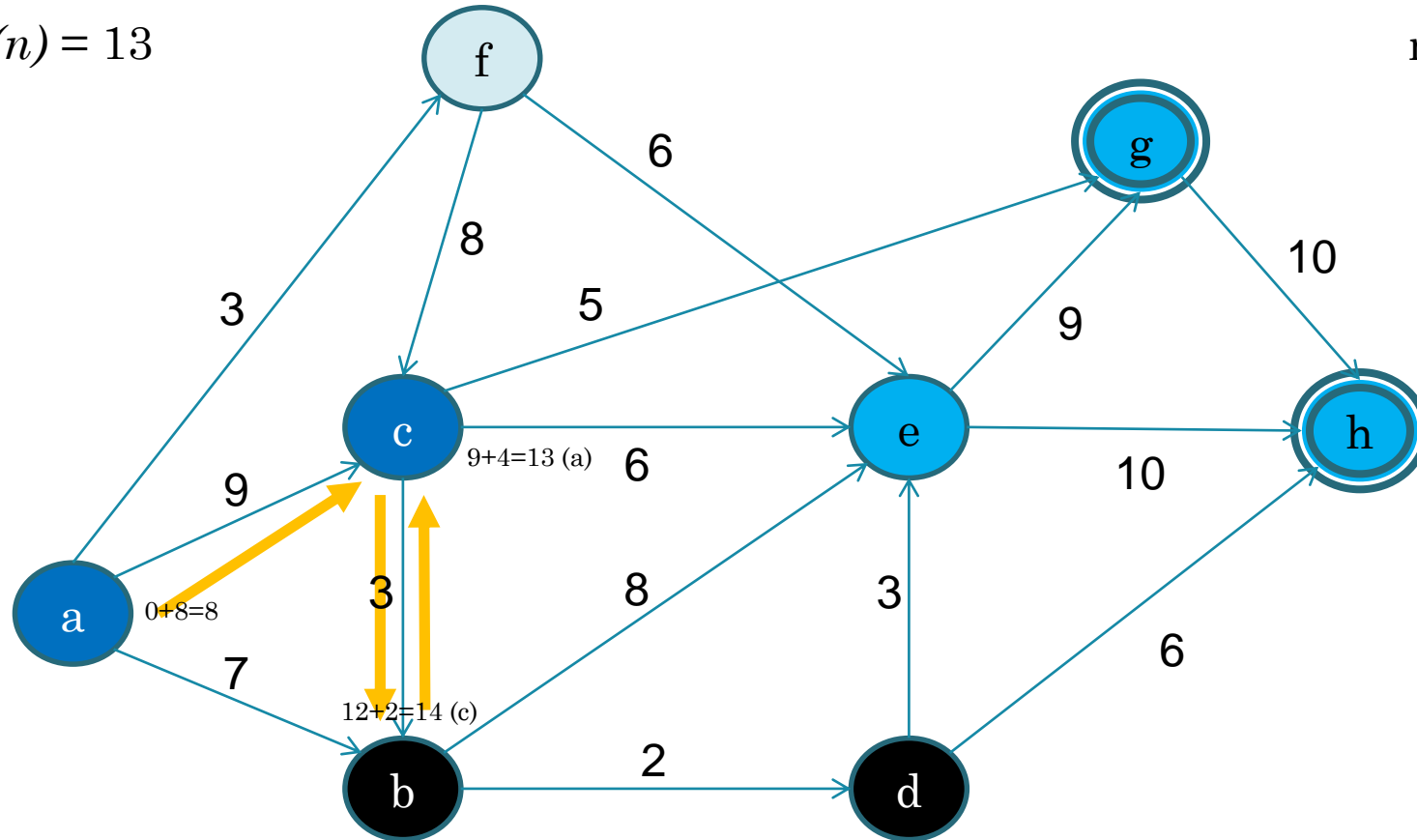


	a	b	c	d	e	f	g	h
$h(n)$	8	2	4	3	9	12	0	0

PRIMER – IDA* (24/36)

Meja $f(n) = 13$

min. čez mejo = 14

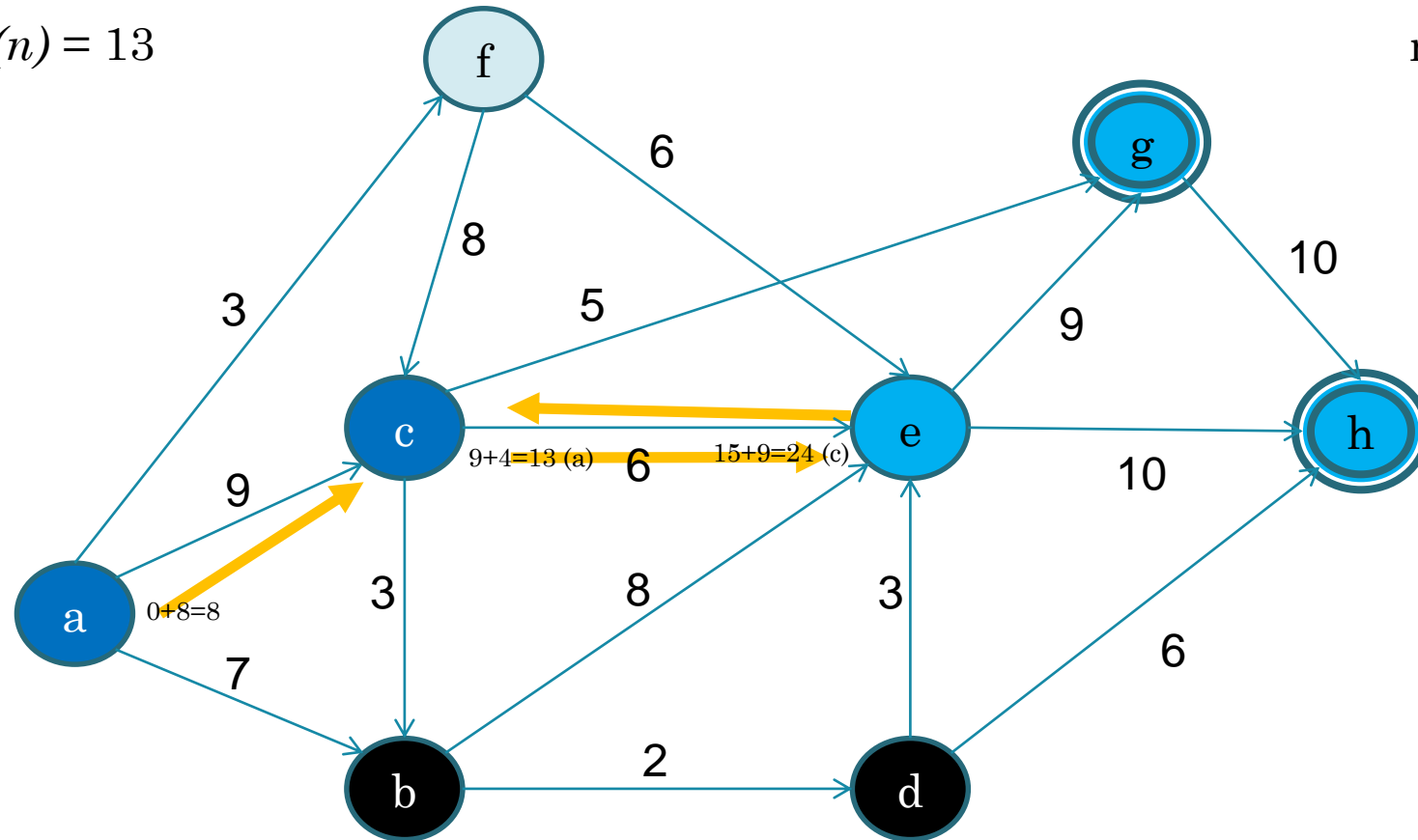


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (25/36)

Meja $f(n) = 13$

min. čez mejo = 14

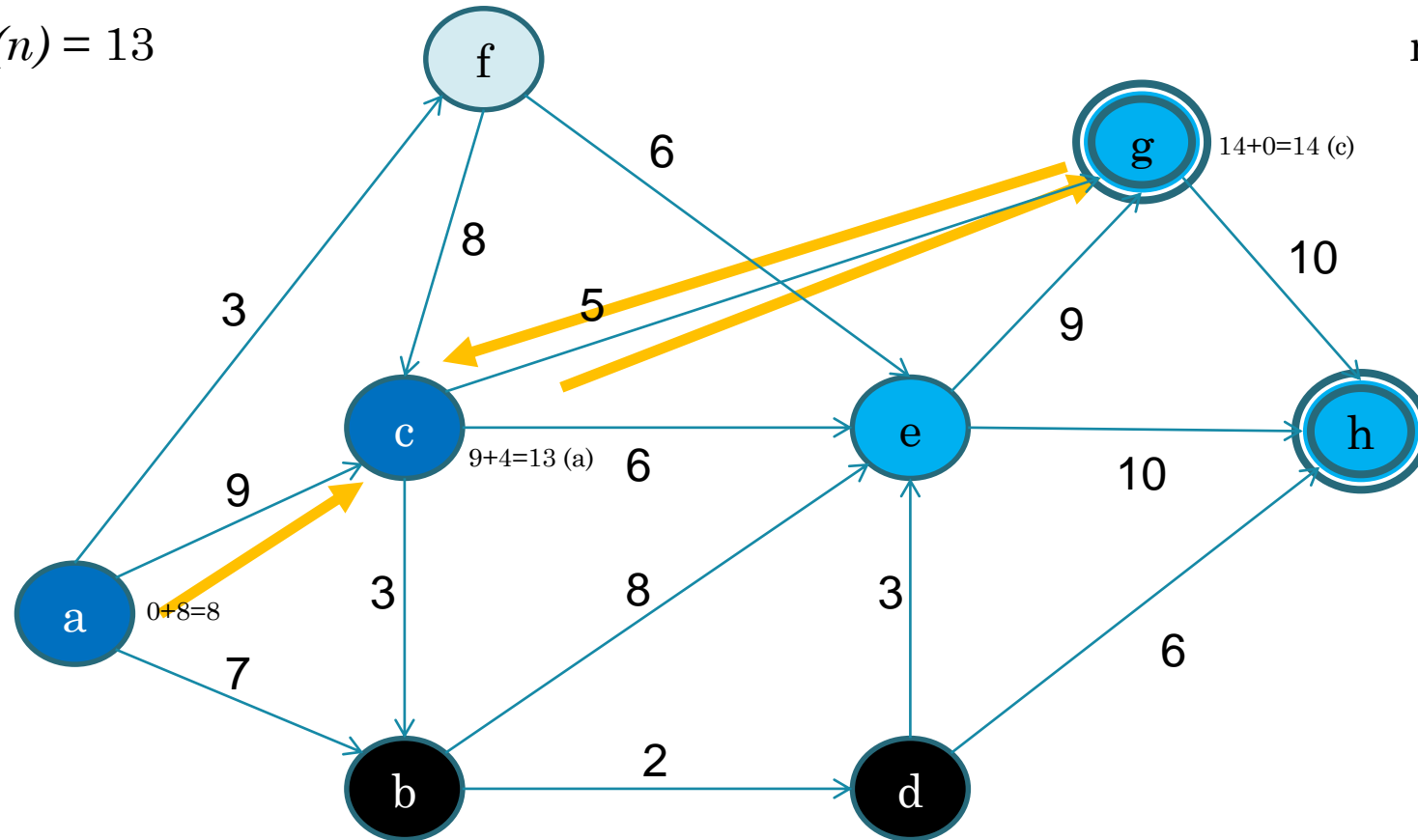


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (26/36)

Meja $f(n) = 13$

min. čez mejo = 14

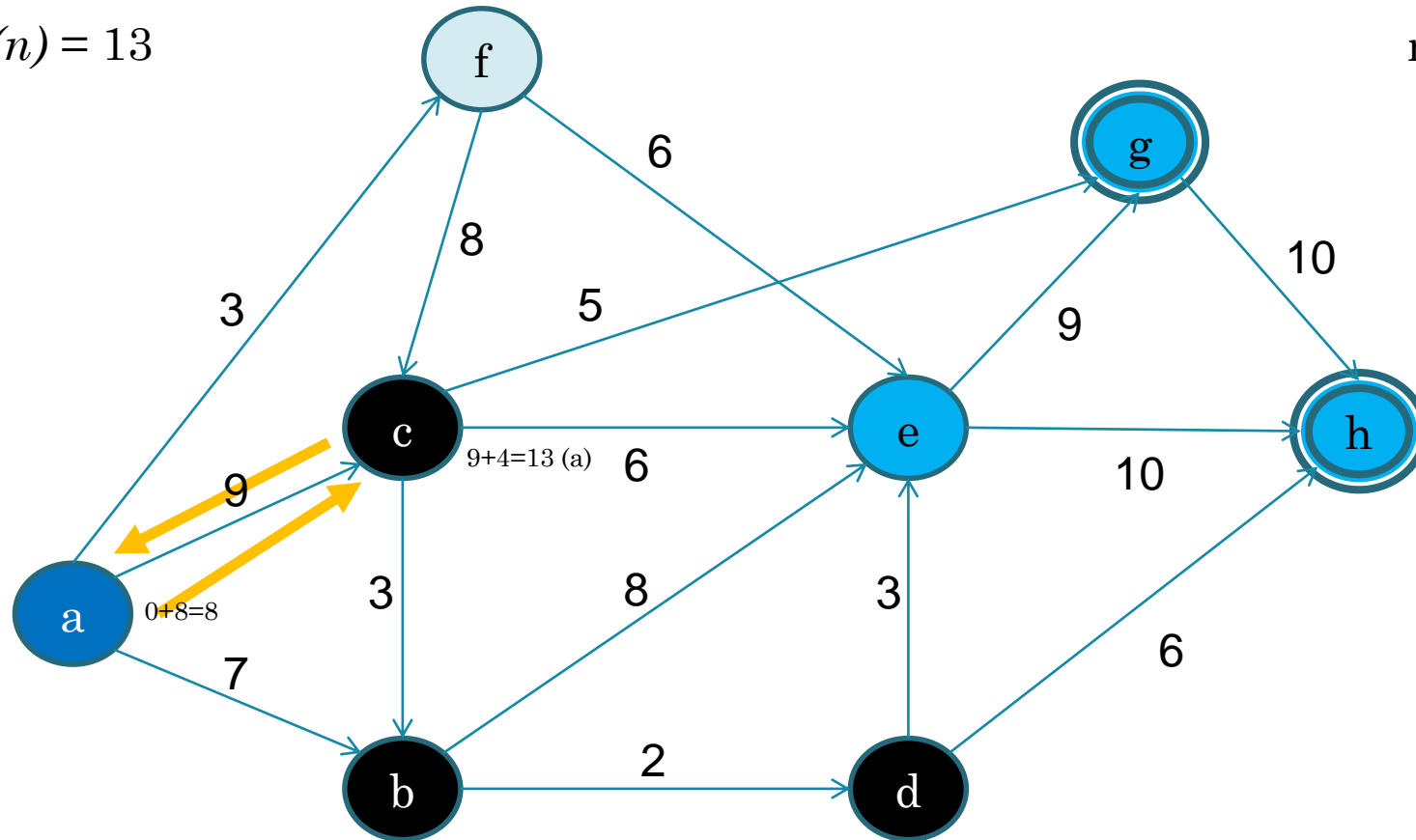


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (27/36)

Meja $f(n) = 13$

min. čez mejo = 14

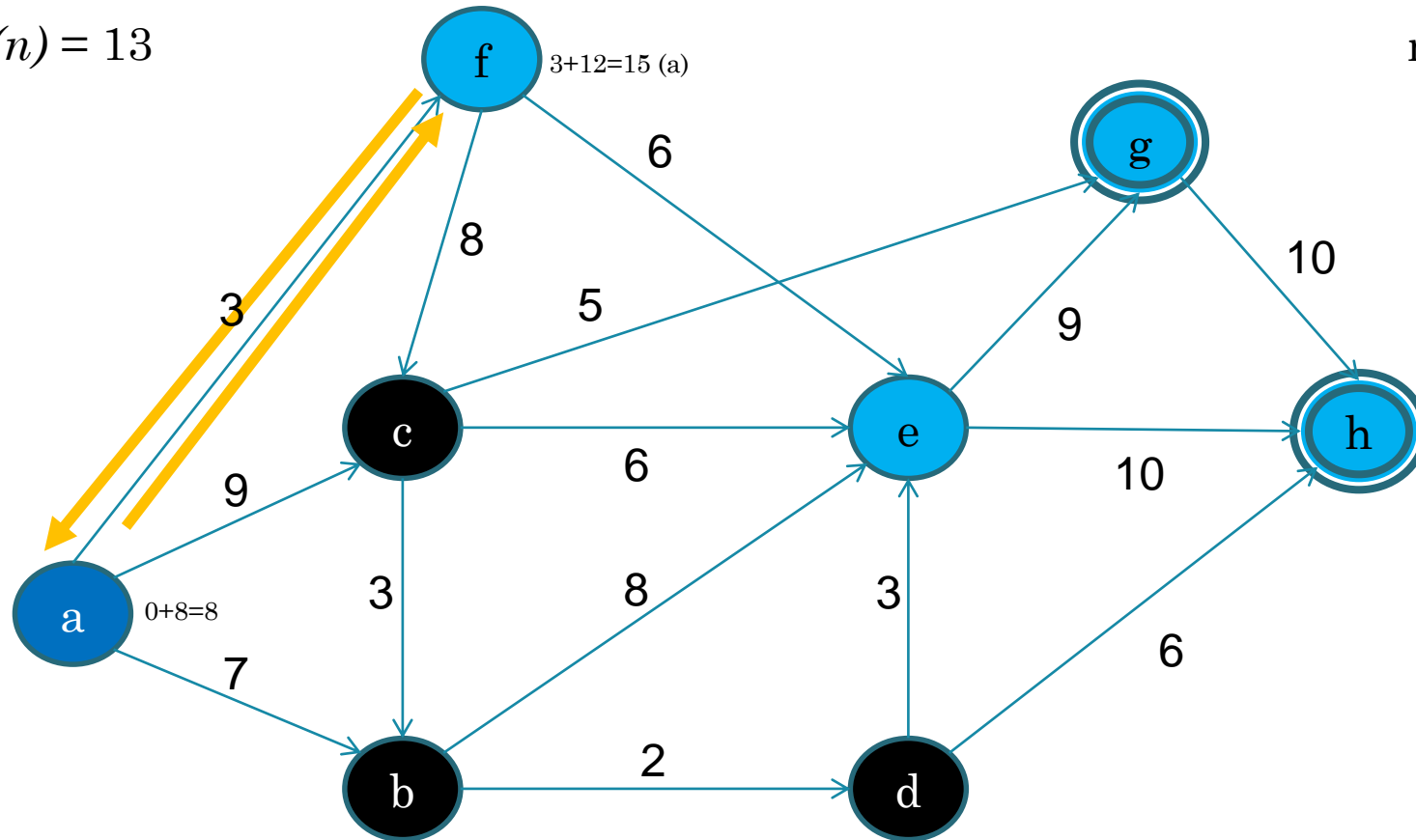


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (28/36)

Meja $f(n) = 13$

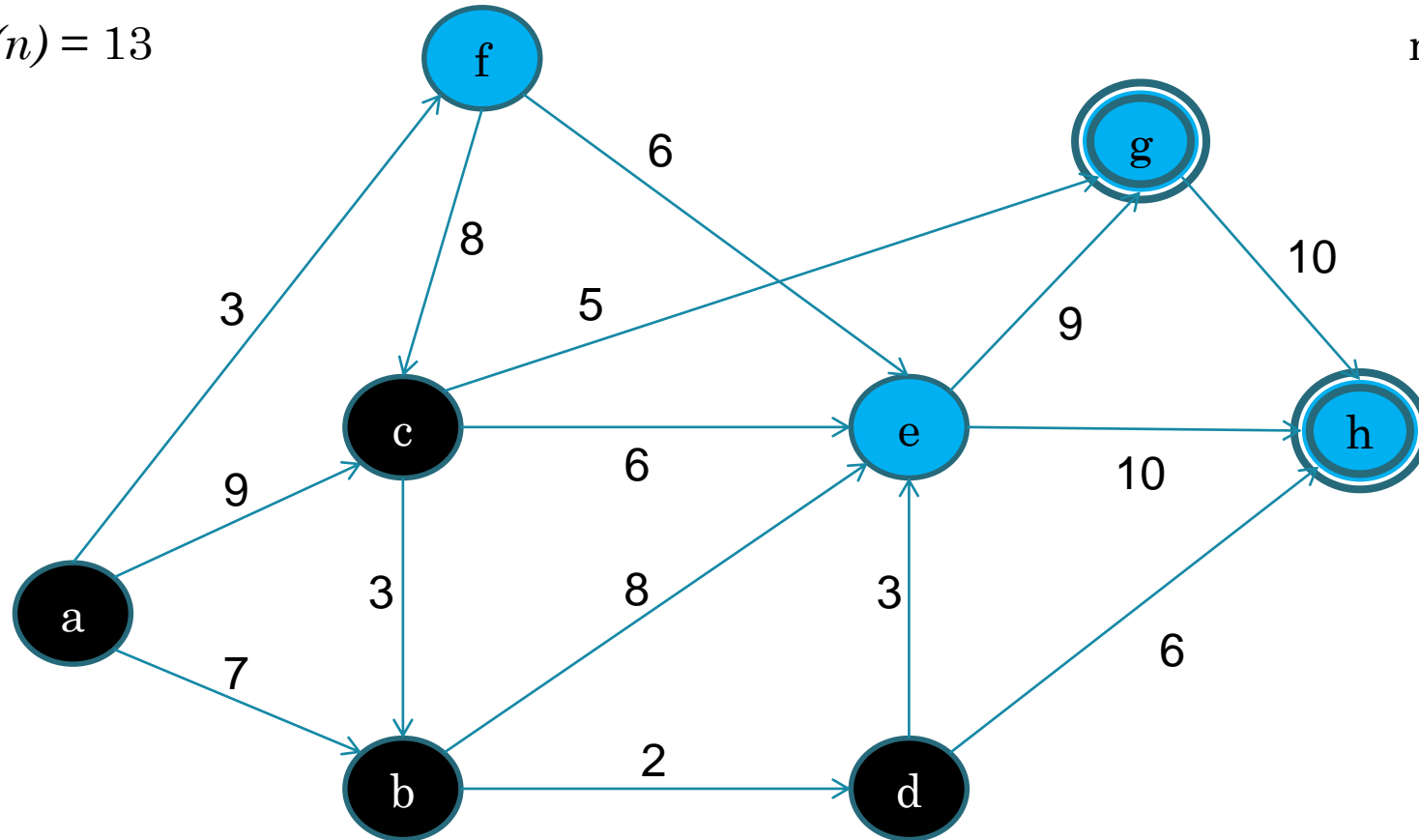
min. čez mejo = 14



	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (29/36)

Meja $f(n) = 13$



min. čez mejo = 14

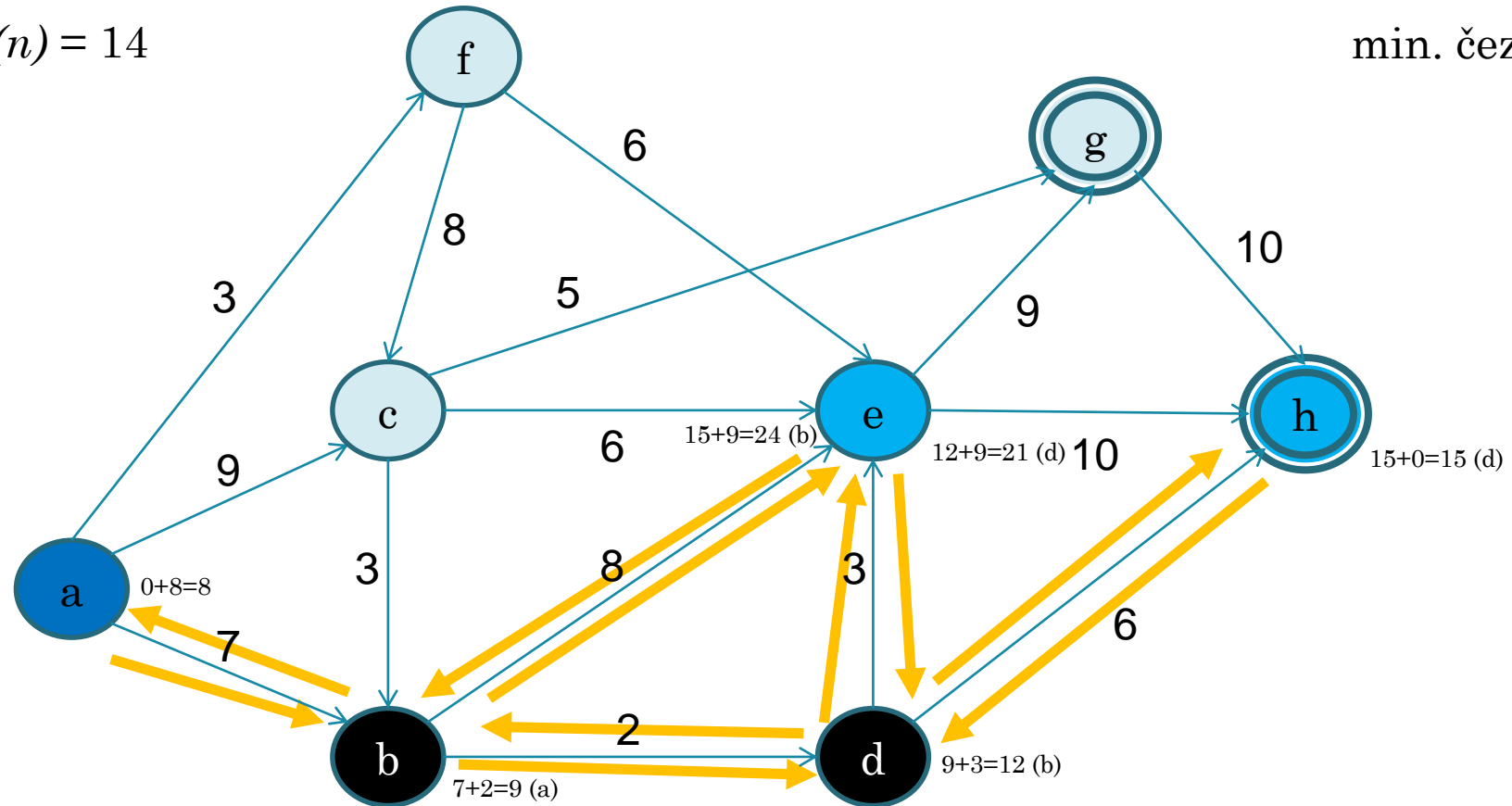
↑
nova meja

	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (30/36)

Meja $f(n) = 14$

min. čez mejo = 15

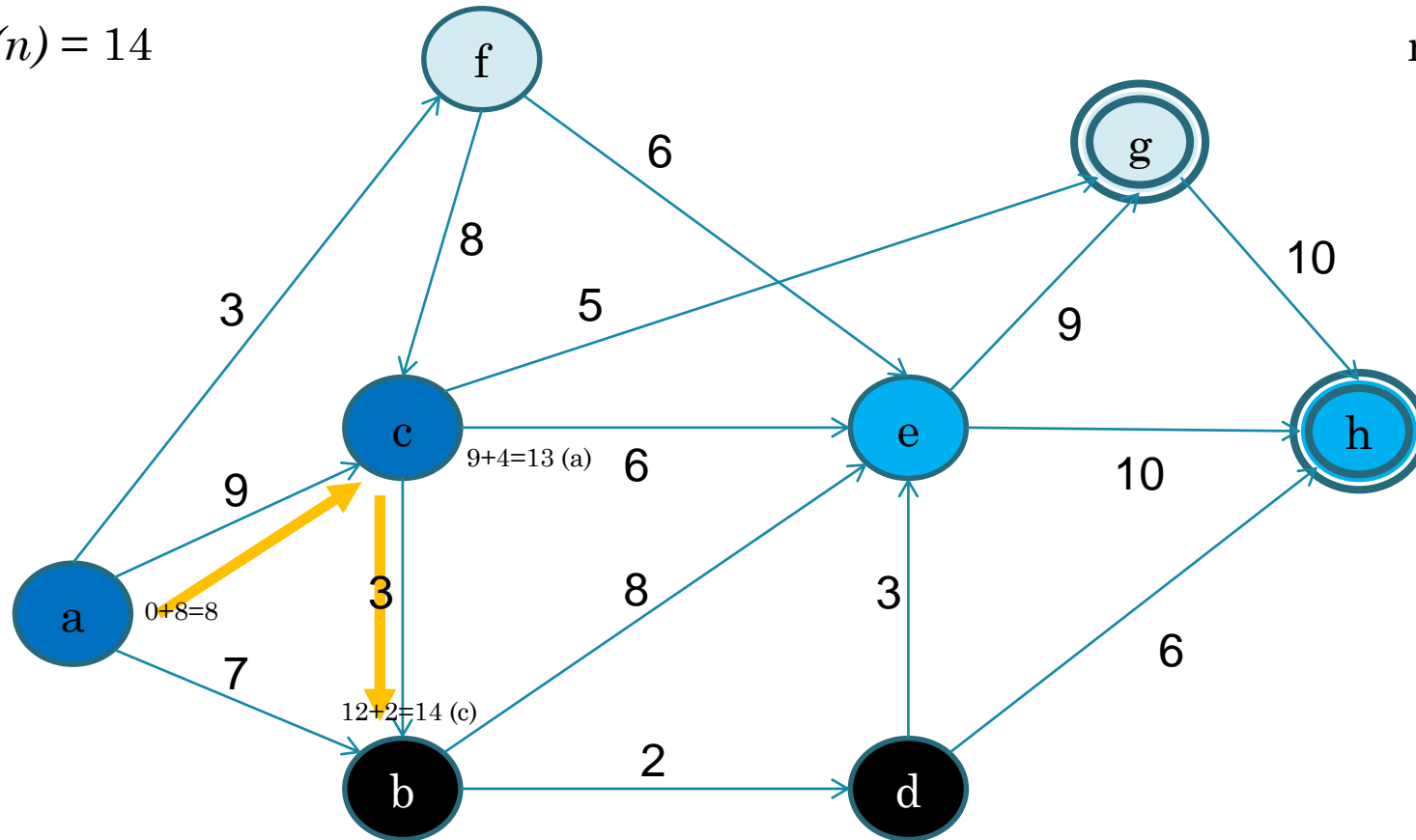


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (31/36)

Meja $f(n) = 14$

min. čez mejo = 15

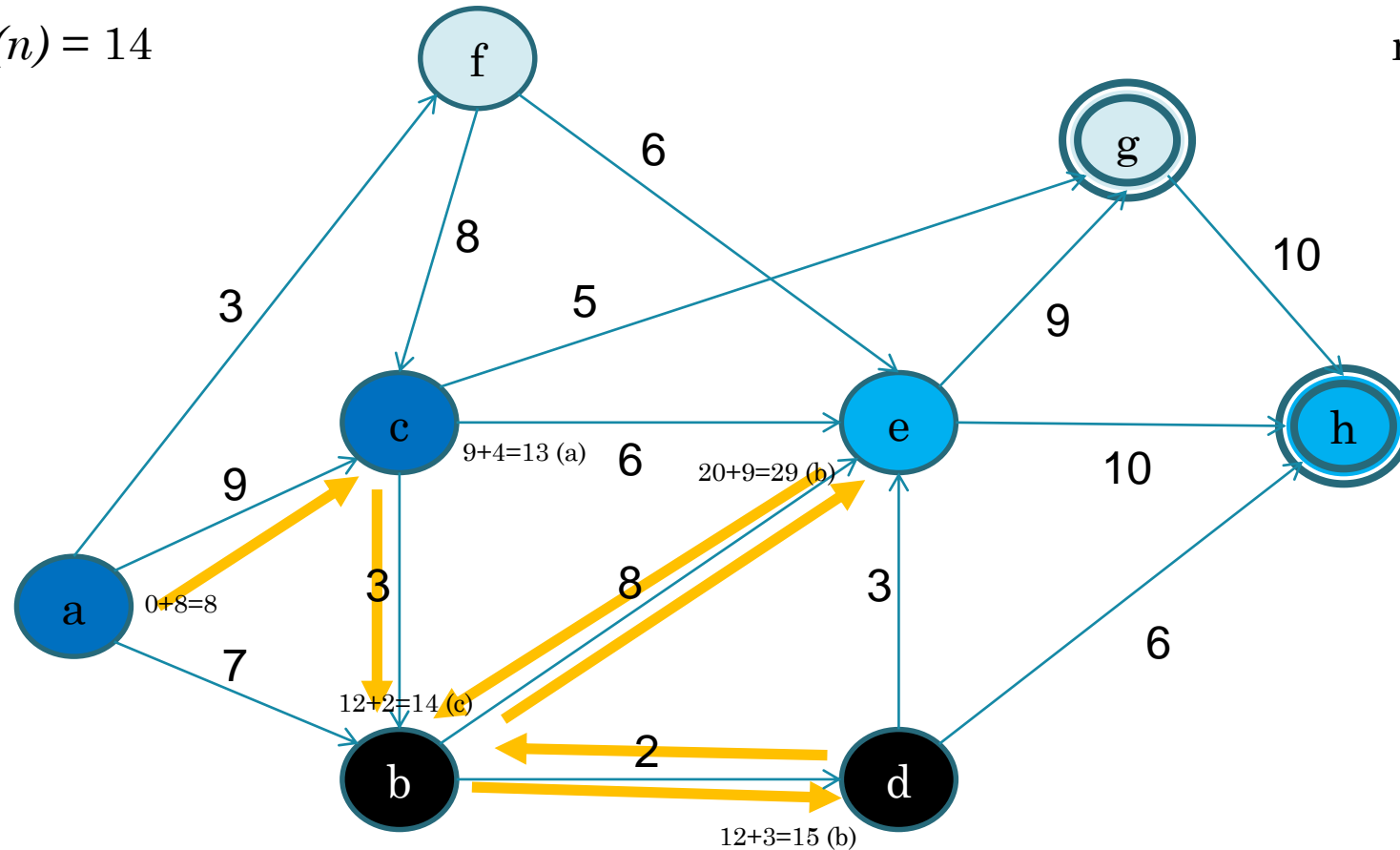


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (32/36)

Meja $f(n) = 14$

min. čez mejo = 15

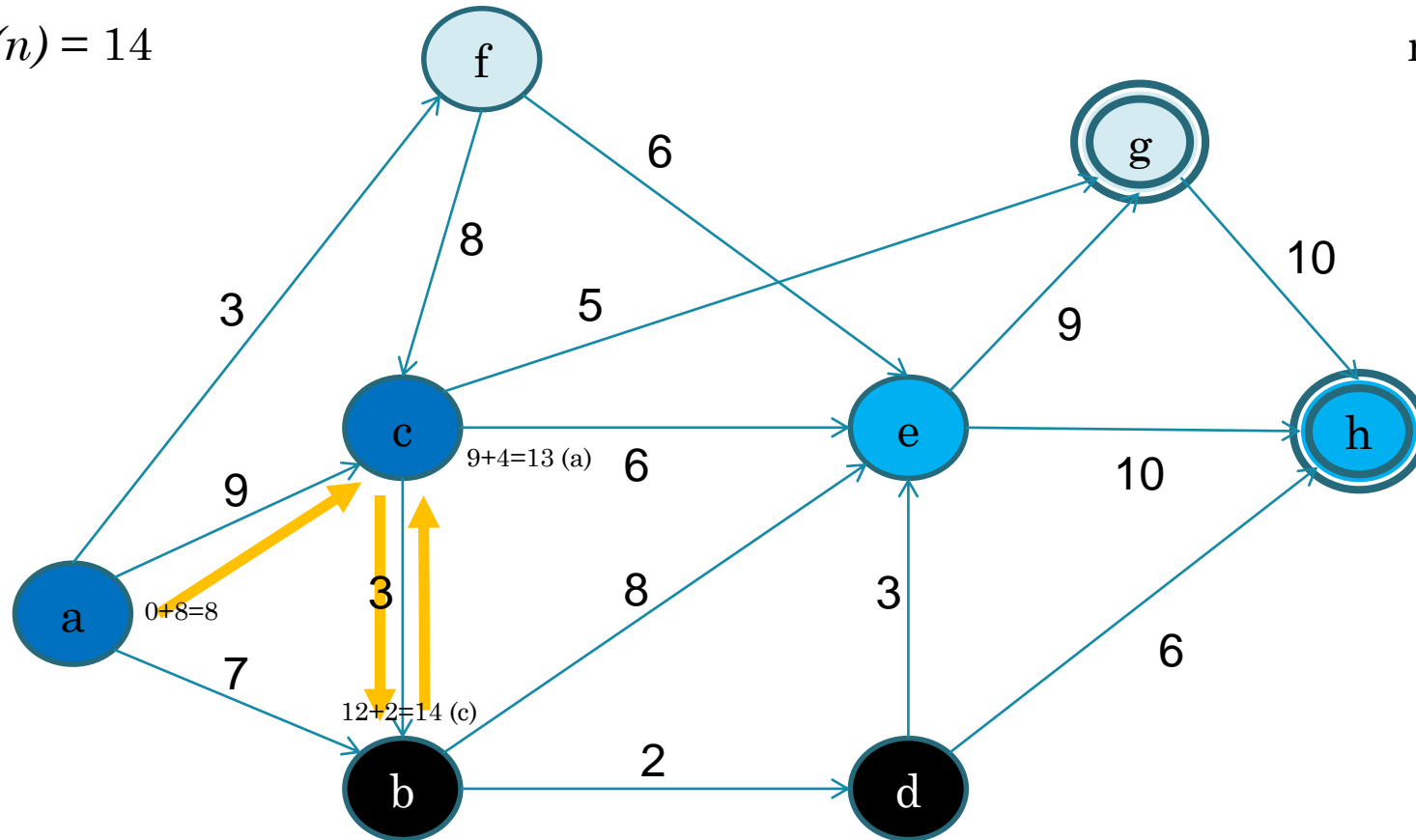


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (33/36)

Meja $f(n) = 14$

min. čez mejo = 15

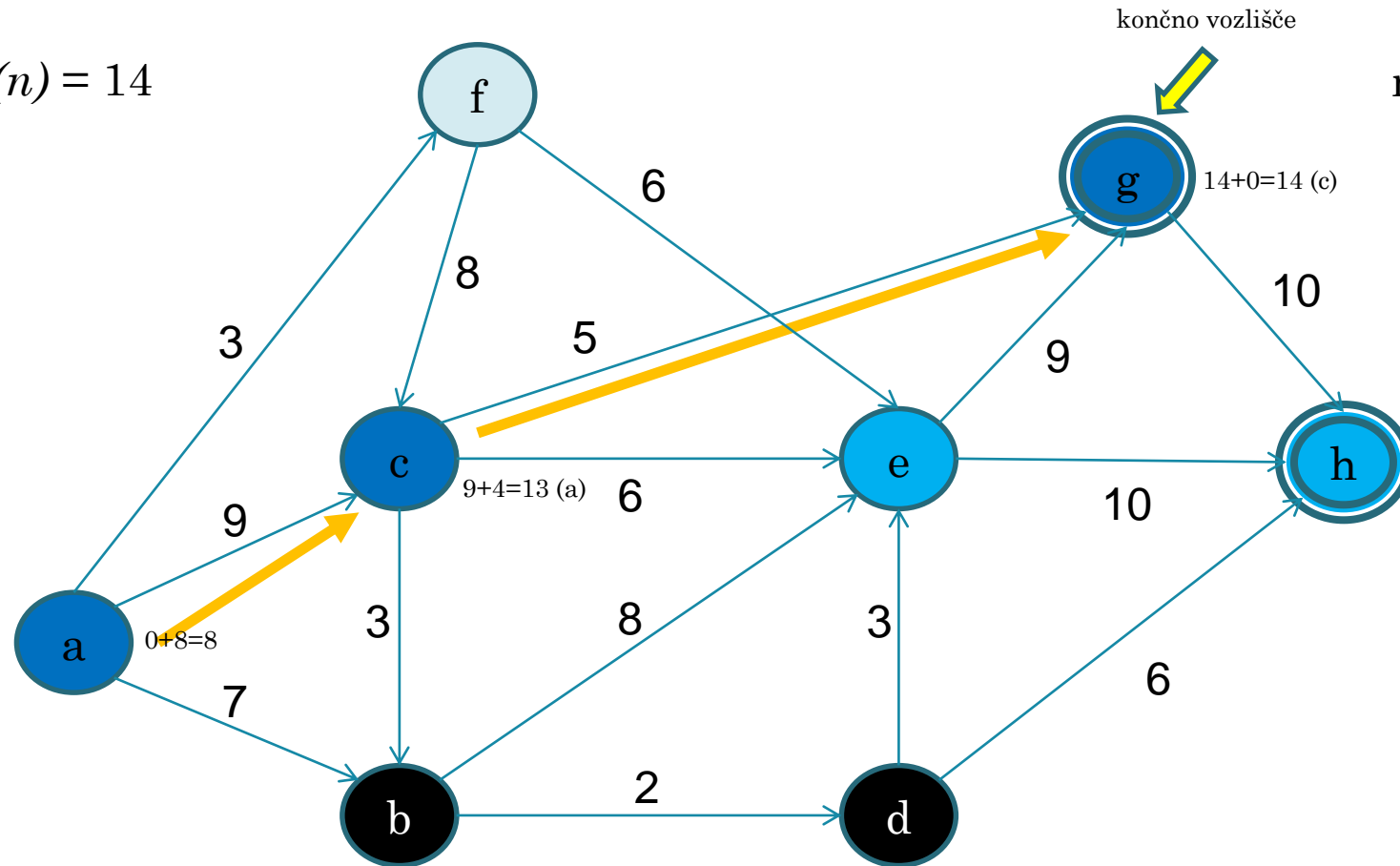


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (35/36)

Meja $f(n) = 14$

min. čez mejo = 15

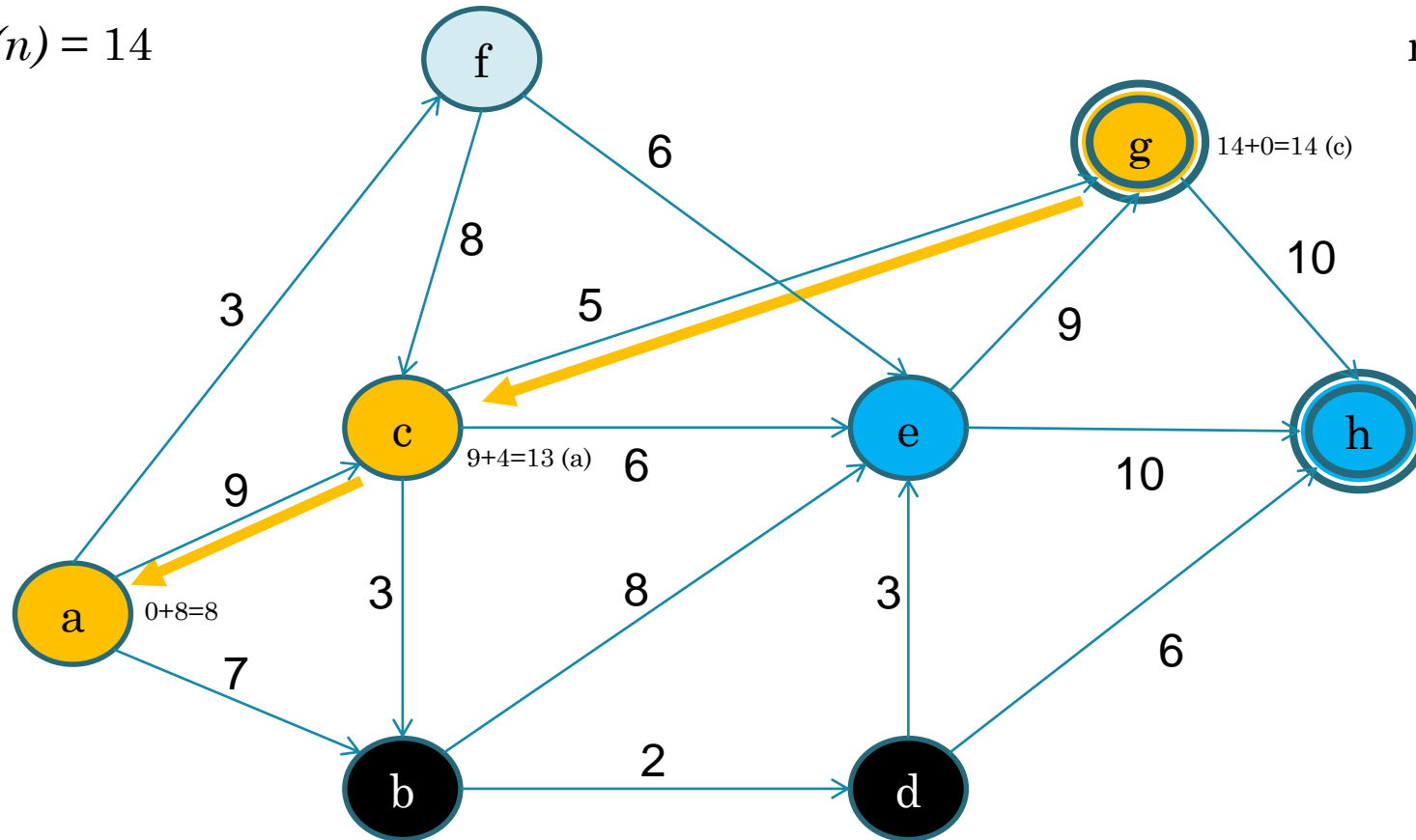


	a	b	c	d	e	f	g	h
h(n)	8	2	4	3	9	12	0	0

PRIMER – IDA* (36/36)

Meja $f(n) = 14$

min. čez mejo = 15



	a	b	c	d	e	f	g	h
$h(n)$	8	2	4	3	9	12	0	0

PRIMER A* – DRSEČE PLOŠČICE (1/8)

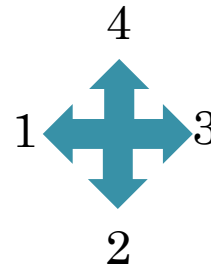
Začetna pozicija:

2	8	3
1	6	4
7		5

Končna pozicija:

1	2	3
8		4
7	6	5

Vrstni red premikov:
(za enakovredne kandidate)



$g(n)$: število potez od začetka

$h(n)$: vsota razdalj (premikov) do pravega mesta
(za vsako ploščico računamo, kot če ostalih ne bi bilo)

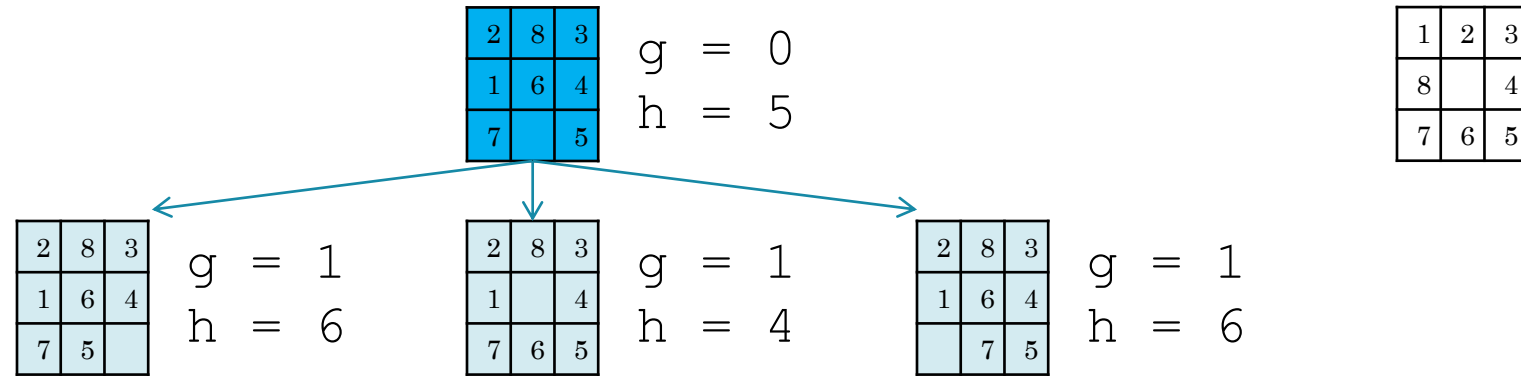
PRIMER A* – DRSEČE PLOŠČICE (2/8)

2	8	3
1	6	4
7		5

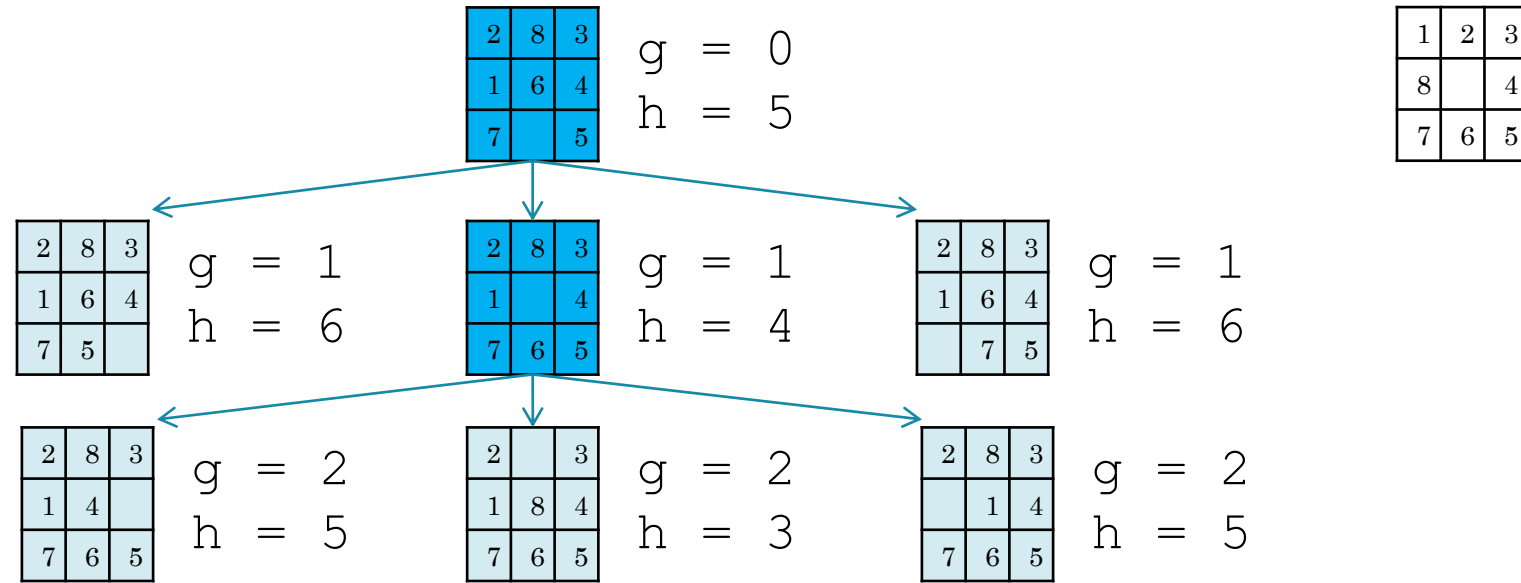
$$g = 0$$
$$h = 5$$

1	2	3
8		4
7	6	5

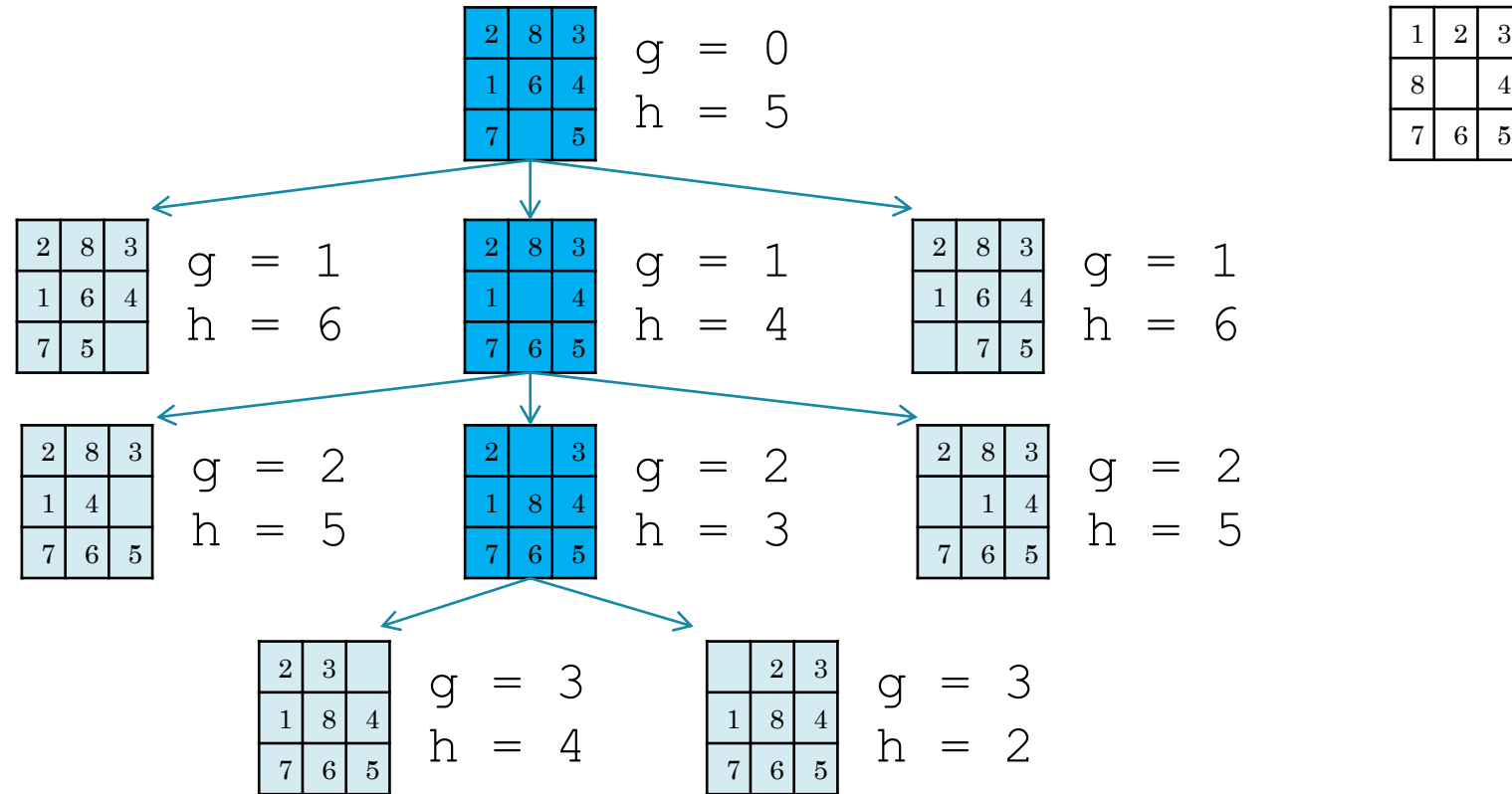
PRIMER A* – DRSEČE PLOŠČICE (3/8)



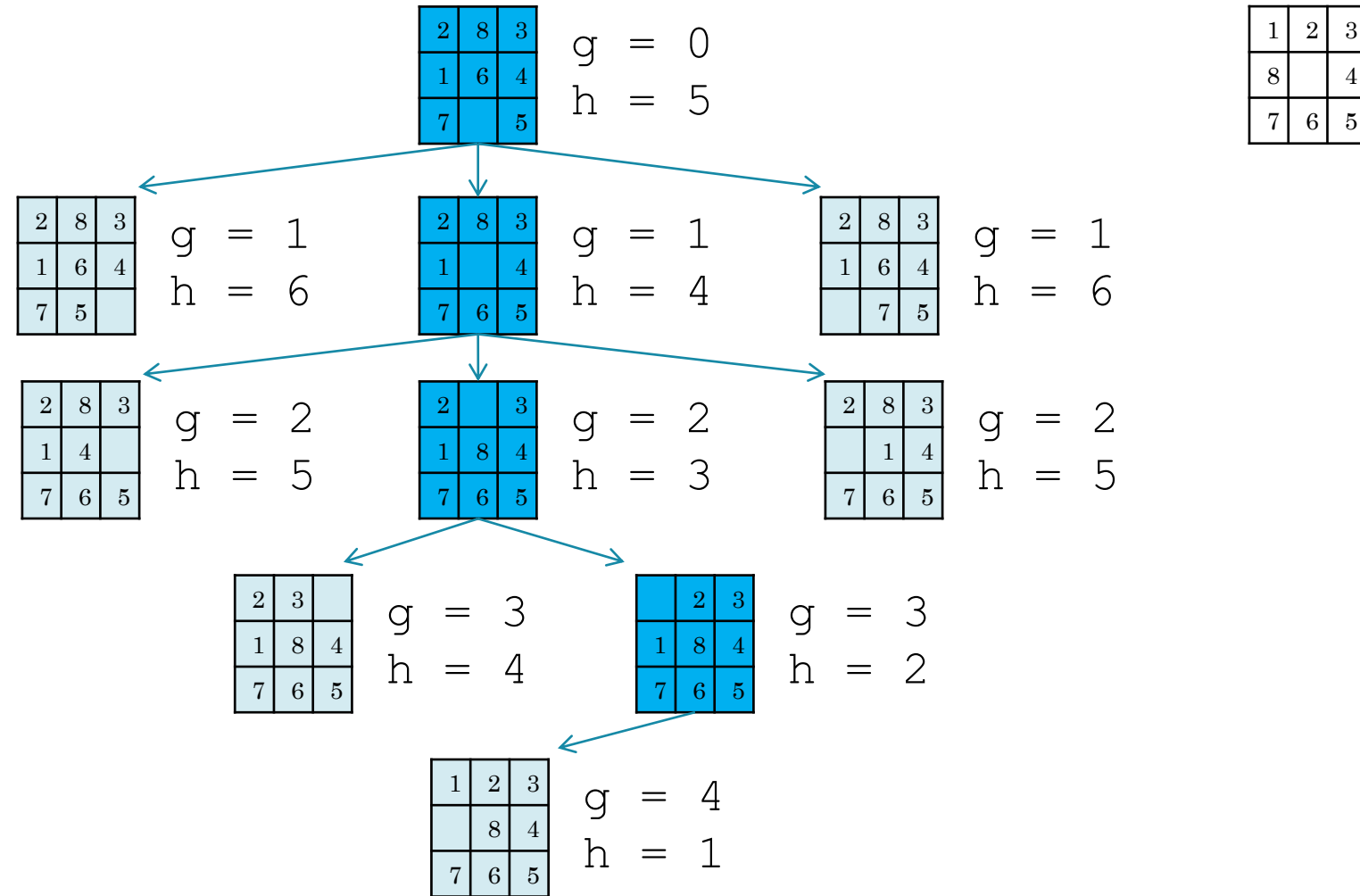
PRIMER A* – DRSEČE PLOŠČICE (4/8)



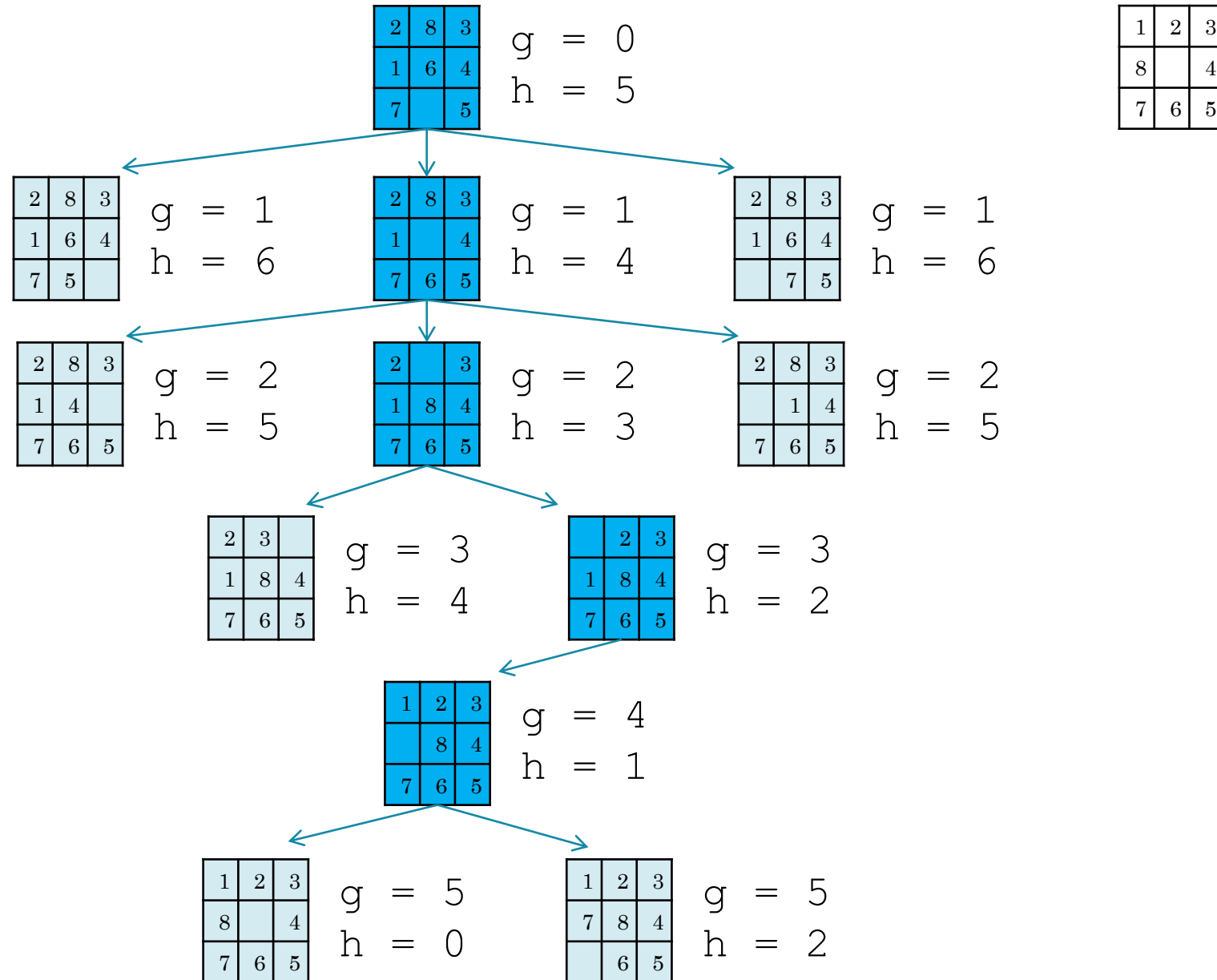
PRIMER A* – DRSEČE PLOŠČICE (5/8)



PRIMER A* – DRSEČE PLOŠČICE (6/8)



PRIMER A* – DRSEČE PLOŠČICE (7/8)



PRIMER A* – DRSEČE PLOŠČICE (8/8)

