Lab9 Solution

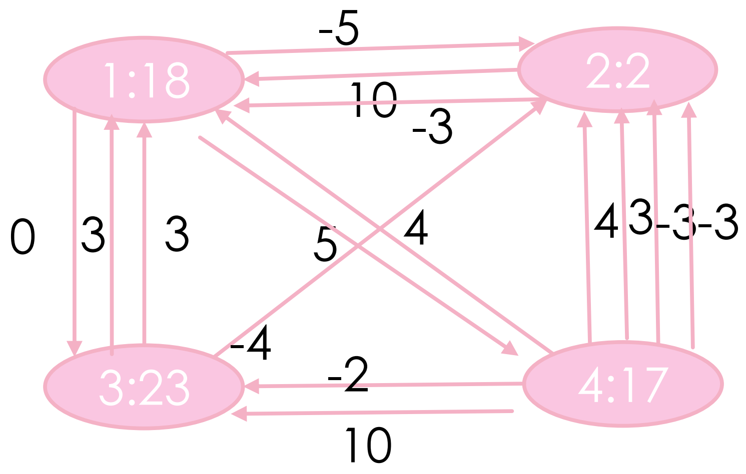
YAO ZHAO

Lab9.A: Voidwalker

- Once there was a voidwalker in the VOID who was eager for power.
- The VOID consists of N spots and M one-way tunnels. The voidwalker can walk through tunnel i by spending w_i energy (a positive w_i means lossing w_i energy, a negative w_i means gaining $|w_i|$ energy, while $w_i = 0$ means no effect).
- The characteristic of energy differs in different spots. In spot i, the voidwalker can compose an energy core using a_i energy or decompose an energy core into a_i energy. The voidwalker can carry at most one energy core with him while walking through a tunnel, since there will be a disaster otherwise.
- The voidwalker begins its journey at spot S with initial power 0. It wonders the maximum energy it can reach with no more than $2K$ operations (an operation is either a compose or a decompose). It is valid for the energy to fall to negative.
- If the voidwalker can gain infinite energy, output **INVINCIBLE**.

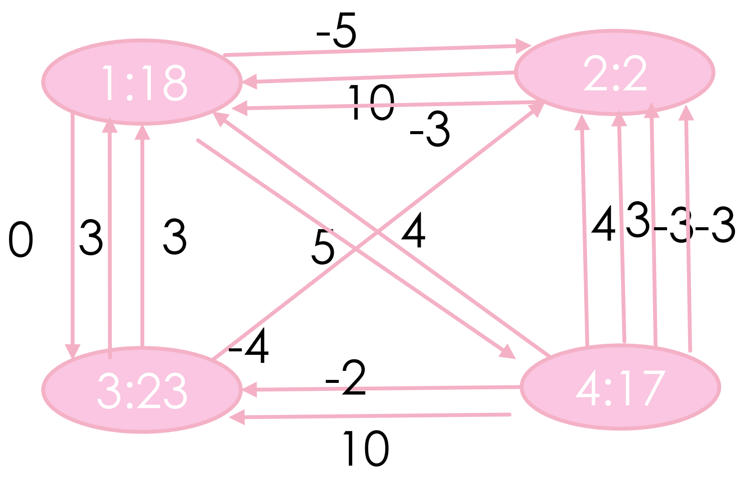
Sample 2 Input

4 15 2 1
18 2 23 17
4 1 4
1 4 5
3 1 3
4 3 10
2 1 10
4 2 3
3 2 -4
3 1 3
1 2 -5
1 3 0
4 2 4
4 2 -3
2 1 -3
4 3 -2
4 2 -3



start from S=1 initial energy E = 0
Easy find there are some cycles which total w_i < 0:
1→2→1:-8
4→2→1→4:-1
3→2→1→3:-7

INVINCIBLE



1	2	3	4
0	∞	∞	∞

bellmanFord iteration 1

1	2	3	4
0	-5	0	5

bellmanFord iteration 2

1	2	3	4
-8	-5	0	5

bellmanFord iteration 3

1	2	3	4
-8	-13	-8	-3

bellmanFord iteration 4

1	2	3	4
-16	-13	-8	-3

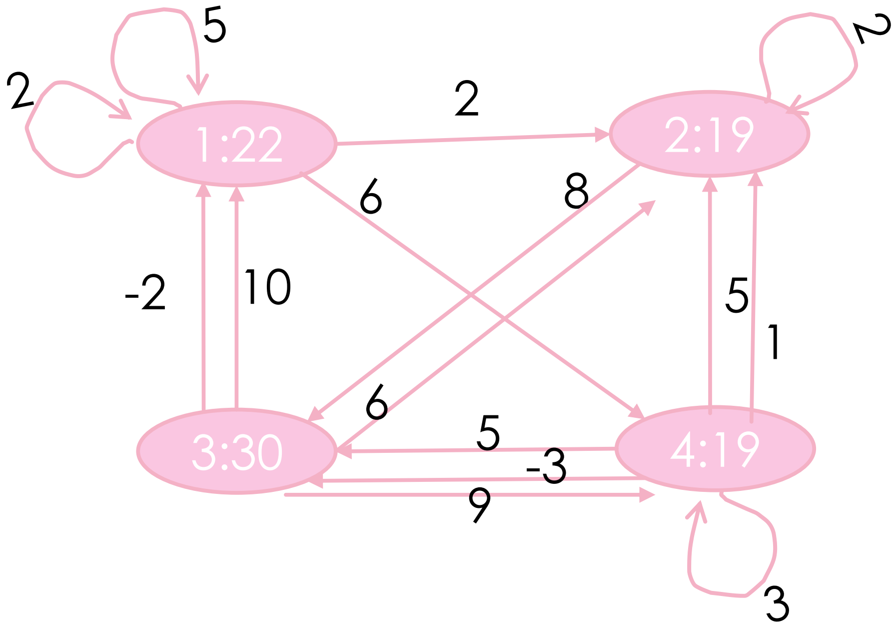
repeat n **iterations**

If some values are still updated in the nth iteration, output "invincible" , stop.
If no values are updated in iteration i≤n,
You can return.

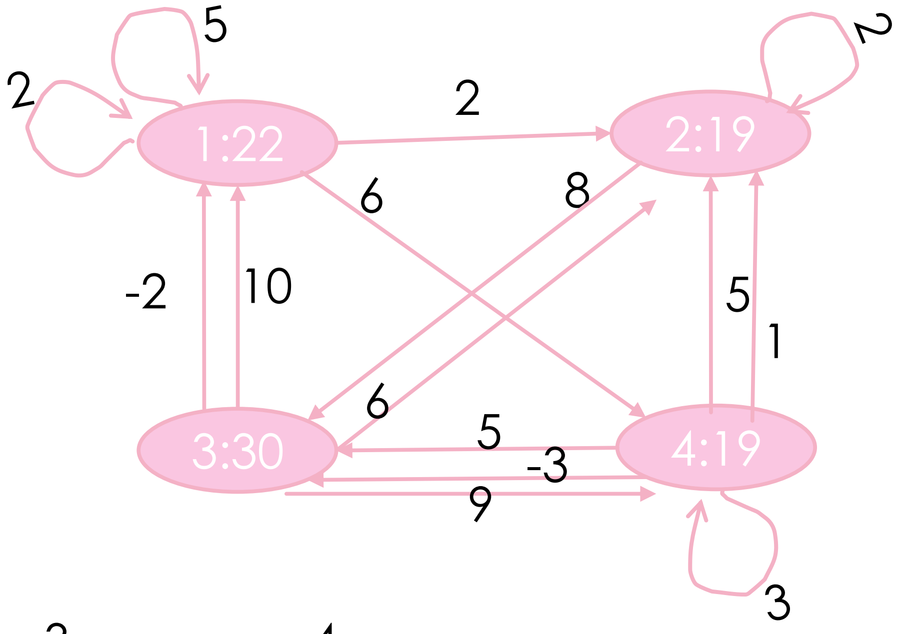
```
Push-Based-Shortest-Path(G, s, t) {
   foreach node v ∈ V {
      M[v] ← ∞
      successor[v] ← ϕ
   }

   M[t] = 0
   for i = 1 to n {
      foreach node w ∈ V {
      if (M[w] has been updated in previous iteration) {
         foreach node v such that (v, w) ∈ E {
            if (M[v] > M[w] + c_vw) {
               M[v] ← M[w] + c_vw
               successor[v] ← w
            }
         }
      }
      If no M[w] value changed in iteration i, stop.
      If i == n and still have M[w] value changed, output "INVINCIBLE"
   }
}
```

Sample 3 Input

4 15 3 3
22 19 30 19
2 3 8
3 2 6
1 2 2
3 1 10
4 3 5
3 4 9
1 1 2
4 2 5
4 4 3
1 1 5
4 3 -3
4 2 1
2 2 2
3 1 -2
1 4 6



using f[0][n] to record the minimum distance from each node to s
1st compose an energy core
let f[1][i] = f[0][i]+a[i]
then do a bellman Ford using f[1][n]
f[1][n] records the minimum cost from s to i and compose an energy core in the midway.
let f[2][i] = f[1][i]-a[i]
then do a bellman Ford using f[2][n]
f[2][n] records the minimum cost from s to i and compose an energy core and decompose an energy in the midway.
repeat until spend all 2K operations.
output the minimum value in f[0][n] ~f[2k][n]



f[0]： initial

1	2	3	4
∞	∞	0	∞

f[0]： final

Bellman Ford

1	2	3	4
-2	0	0	4

+a[i]

f[1]： initial

1	2	3	4
20	19	30	23

Bellman Ford

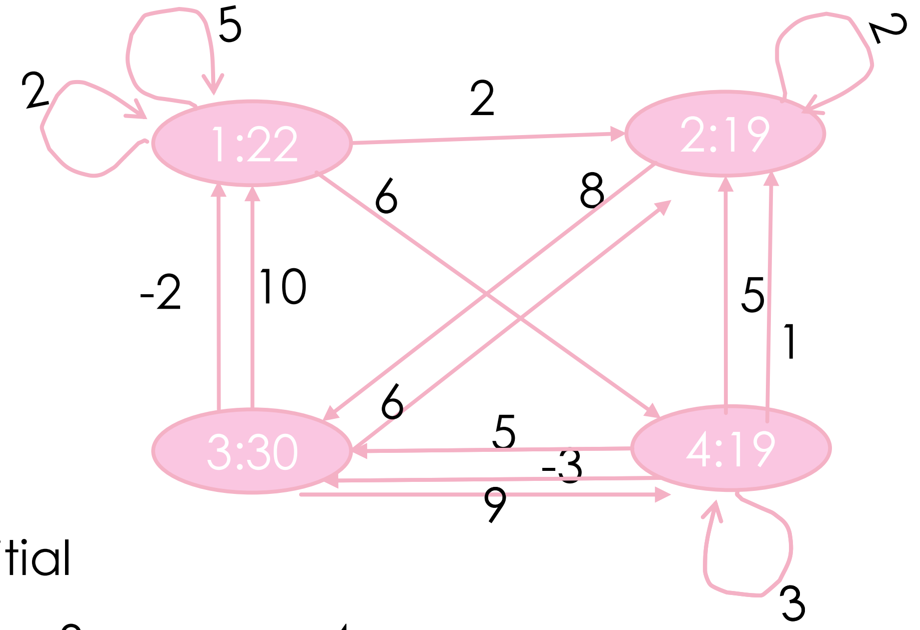
f[1]： final

1	2	3	4
18	19	20	23

-a[i]

f[2]： initial

1	2	3	4
-4	0	-10	4



f[2]:initial

1	2	3	4
-4	0	-10	4

f[2]:final Bellman Ford

1	2	3	4
-12	-10	-10	-6

+a[i]

f[3]:initial

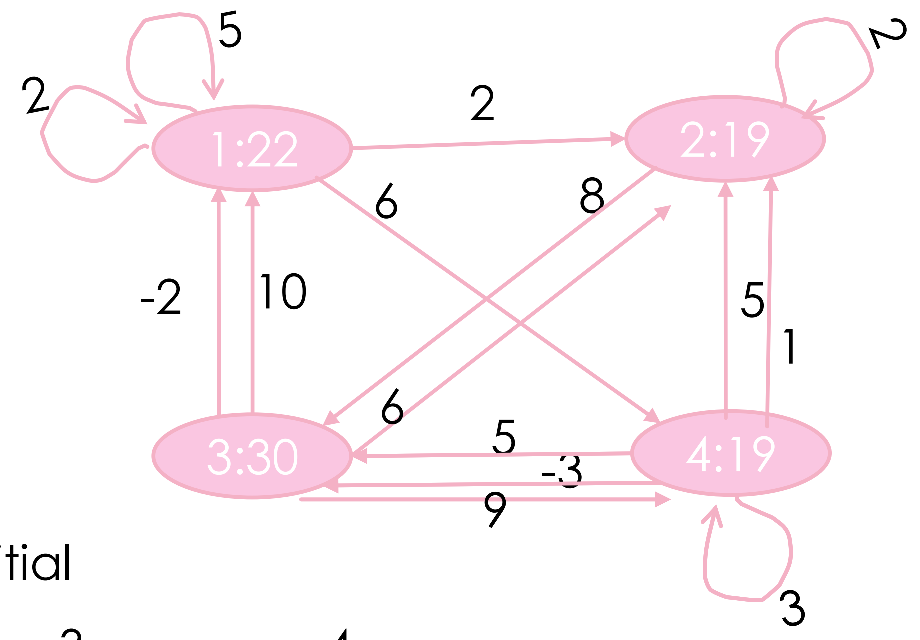
1	2	3	4
10	9	20	13

f[3]:final Bellman Ford

1	2	3	4
8	9	10	13

f[4]： initial -a[i]

1	2	3	4
-14	-10	-20	-6



f[4]:initial

1	2	3	4
-14	-10	-20	-6

f[4]:final ↓ Bellman Ford

1	2	3	4
-22	-20	-20	-16

+a[i]

f[5]:initial

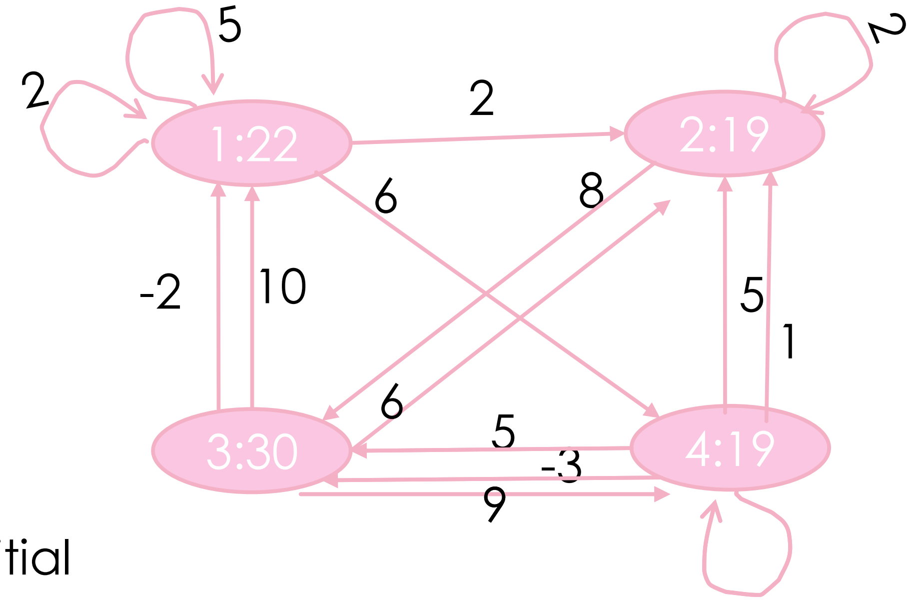
1	2	3	4
0	-1	10	3

f[5]:final ↓ Bellman Ford

1	2	3	4
-2	0	0	3

f[6]:initial ↓ -a[i]

1	2	3	4
-24	-19	-30	-16



f[6]:initial

1	2	3	4
-24	-19	-30	-16

f[6]:final

Bellman Ford

1	2	3	4
-32	-30	-30	-26

Sample 3 Output

32

Lab9.B: Scream Out Loud

- Lida Pu has long suffered from a compulsion to obtain symmetric things, for example, palindrome strings.
- One day, Lida Pu received a secret mail, in which he saw a string template. The template contains lowercase letters, symbol '?' corresponding to an arbitrary letter and symbol '*' corresponding to a zero or more arbitrary letters.
- Please tell Lida Pu the minimum length of the palindrome string which can be obtained from the given template. If he cannot get a palindrome string anyhow, just tell him to face the reality.
- Note that in Lida Pu's mind, an empty string is also a palindrome string.

symbol '?' corresponding to an arbitrary letter

symbol '*' corresponding to a zero or more arbitrary letters

Sample 1 Input

***ac?ba**

try length = 5 → **ac?ba** **fail**

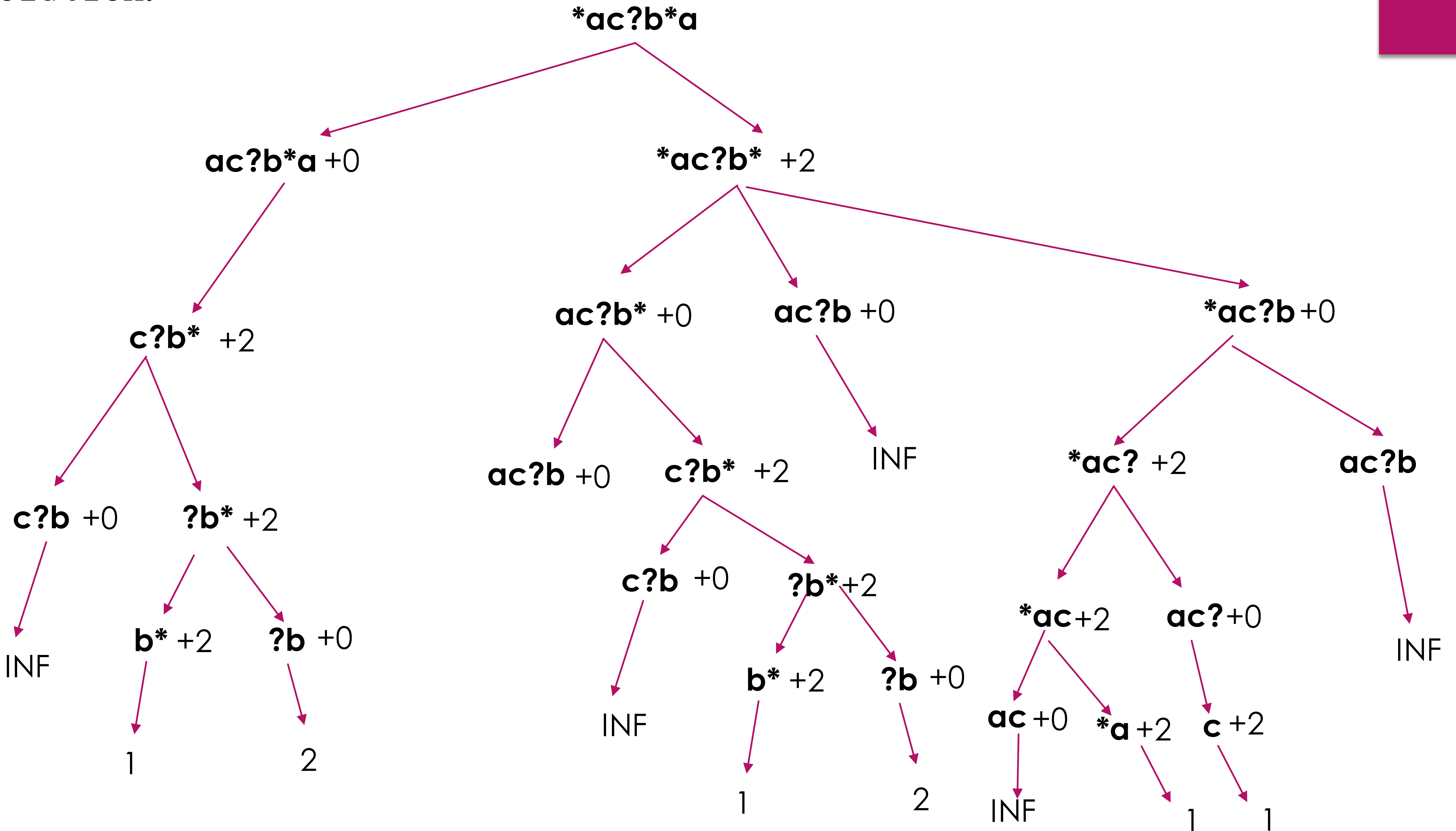
try length = 6 $\xrightarrow{\text{let * = a}}$ **aac?ba** **fail**

try length = 7 $\xrightarrow{\text{let * = ab}}$ **abac?ba** $\xrightarrow{\text{let ? = a}}$ **abacaba** **success!**

↓

7

1. Characterize the structure of an optimal solution.



2. Recursively define the value of an optimal solution.

let |s| = n
Input String: c[1]c[2]…c[n]
Let opt[i][j] is the **minimum** length of the palindrome string c[i]c[i+1]…c[j]， then：
if c[i] == ‘*’
opt[i][j] = min{opt[i+1][k]+0, opt[i][j-1]+2}
if c[j] == ‘*’

____

if c[i] == ‘*’&&c[j]==‘*’

____

if c[i] can match c[j]
opt[i][j] = min{opt[i+1][j-1]}

3. Compute the value of an optimal solution, typically in a bottom-up fashion.

***ac?b*a**

	1	2	3	4	5	6	7
1	0	1	3	3	5	5	6
2		1	∞	3	∞	6	6
3			1	2	∞	4	∞
4				1	2	2	3
5					1	1	∞
6						0	1
7							1

acbbca