GSP (Generalised Sequential Patterns)

SID	Sequence
1	$\langle \{a, b\}, \{c\}, \{f, g\}, \{g\}, \{e\} \rangle$
2	$\langle \{a, d\}, \{c\}, \{b\}, \{a, b, e, f\} \rangle$
3	$\langle \{a\}, \{b\}, \{f, g\}, \{e\} \rangle$
4	$\langle \{b\}, \{f, g\} \rangle$

- Min sup =2

1- length Seq	Support
{a}	3
{b}	4
{c}	2
{d}	1
{e}	3
{f}	4
{g}	3

Candidates are generated in two steps:

1. **Join Phase.** We generate candidate sequences by joining L_{k-1} with L_{k-1}. A sequence s_1 joins with s_2 if the subsequence obtained by dropping the first item of s_1 is the same as the subsequence obtained by dropping the last item of s_2. The candidate sequence generated by joining s_1 with s_2 is the sequence s_1 extended with the last item in s_2. The added item becomes a separate element if it was a separate element in s_2, and part of the last element of s_1 otherwise. When joining L_1 with L_1, we need to add the item in s_2 both as part of an itemset and as a separate element, since both $\langle$(x) (y)$\rangle$ and $\langle$(x y)$\rangle$ give the same sequence $\langle$(y)$\rangle$ upon deleting the first item. (Observe that s_1 and s_2 are contiguous subsequences of the new candidate sequence.)
2. **Prune Phase.** We delete candidate sequences that have a contiguous $(k-1)$-subsequence whose support count is less than the minimum support. If there is no max-gap constraint, we also delete candidate sequences that have any subsequence without minimum support.

SID	Sequence
1	$\langle \{a, b\}, \{c\}, \{f, g\}, \{g\}, \{e\} \rangle$
2	$\langle \{a, d\}, \{c\}, \{b\}, \{a, b, e, f\} \rangle$
3	$\langle \{a\}, \{b\}, \{f, g\}, \{e\} \rangle$
4	$\langle \{b\}, \{f, g\} \rangle$

- {ab} not equal to {ba}
- {(a,b)} equal to {(b,a)}

1- length Seq	Support
{a}	3
{b}	4
{c}	2
{e}	3
{f}	4
{g}	3

2-len Seq (a)	
{ab}	2
{ac}	2
{ae}	3
{af}	3
{ag}	2
{(a,b)}	2
{(a,c)}	0
{(a,e)}	1
{(a,f)}	1
{(a,g)}	0

1- length Seq	support
{a}	3
{b}	4
{c}	2
{e}	3
{f}	4
{g}	3

2-len Seq (a)	2-len Seq (b)	2-len Seq (c)	2-len Seq (e)	2-len Seq (f)	2-len Seq (g)
{ab}	{ba}	{ca}	{ea}	{fa}	{ga}
{ac}	{bc}	{cb}	{eb}	{fb}	{gb}
{ae}	{be}	{ce}	{ec}	{fc}	{gc}
{af}	{bf}	{cf}	{ef}	{fe}	{ge}
{ag}	{bg}	{cg}	{eg}	{fg}	{gf}
{(a,b)}	{(b,c)}	{(c,e)}	{(e,f)}	{(e,g)}	
{(a,c)}	{(b,e)}	{(c,f)}	{(e,g)}		
{(a,e)}	{(b,f)}	{(c,g)}			
{(a,f)}	{(b,g)}				
{(a,g)}					

2-len Seq (a)		2-len Seq (b)		2-len Seq (c)		2-len Seq (e)		2-len Seq (f)		2-len Seq (g)	
{ab}	2	{ba}	1	{ca}	1	{ea}	0	{fa}	0	{ga}	0
{ac}	2	{bc}	1	{cb}	1	{eb}	0	{fb}	0	{gb}	0
{ae}	3	{be}	3	{ce}	2	{ec}	0	{fc}	0	{gc}	0
{af}	3	{bf}	4	{cf}	2	{ef}	0	{fe}	2	{ge}	2
{ag}	2	{bg}	3	{cg}	1	{eg}	0	{fg}	1	{gf}	0
{(a,b)}	2	{(b,c)}	0	{(c,e)}	0	{(e,f)}	1	{(f,g)}	3		
{(a,c)}	0	{(b,e)}	1	{(c,f)}	0	{(e,g)}	1				
{(a,e)}	1	{(b,f)}	1	{(c,g)}	0						
{(a,f)}	1	{(b,g)}	1								
{(a,g)}	0										

Candidates are generated in two steps:

1. **Join Phase.** We generate candidate sequences by joining L_{k-1} with L_{k-1}. A sequence s_1 joins with s_2 if the subsequence obtained by dropping the first item of s_1 is the same as the subsequence obtained by dropping the last item of s_2. The candidate sequence generated by joining s_1 with s_2 is the sequence s_1 extended with the last item in s_2. The added item becomes a separate element if it was a separate element in s_2, and part of the last element of s_1 otherwise. When joining L_1 with L_1, we need to add the item in s_2 both as part of an itemset and as a separate element, since both $\langle$ (x) (y) $\rangle$ and $\langle$ (x y) $\rangle$ give the same sequence $\langle$ (y) $\rangle$ upon deleting the first item. (Observe that s_1 and s_2 are contiguous subsequences of the new candidate sequence.)
2. **Prune Phase.** We delete candidate sequences that have a contiguous (k–1)-subsequence whose support count is less than the minimum support. If there is no max-gap constraint, we also delete candidate sequences that have any subsequence without minimum support.

2-len Seq	
{ab}	2
{ac}	2
{ae}	3
{af}	3
{ag}	2
{(a,b)}	2
{be}	3
{bf}	4
{bg}	3
{ce}	2
{cf}	2
{fe}	2
{(f,g)}	3
{ge}	2

3-len Seq	
{abe}	{(ab)f}
{abf}	{(ab)g}
{abg}	{bfe}
{ace}	{b(f,g)}
{acf}	{bge}
{afe}	{cfe}
{a(f,g)}	{c(f,g)}
{age}	{(f,g)e}
{(ab)e}	

- {ab} {ae} {be}
- {cf} {cg} {(f,g)}

- {ab} {ac}
- {ab} {be} → {abe}
- {(a,b)} {be} → {(a,b)e}
- {bf} { (f,g) } → { b(f,g) }

SID	Sequence
1	$\langle\{a,b\},\{c\},\{f,g\},\{g\},\{e\}\rangle$
2	$\langle\{a,d\},\{c\},\{b\},\{a,b,e,f\}\rangle$
3	$\langle\{a\},\{b\},\{f,g\},\{e\}$
4	$\langle\{b\},\{f,g\}\rangle$

3-len Seq			
{abe}	2	{(ab)e}	1
{abf}	2	{(ab)f}	1
{abg}	1	{(ab)g}	1
{ace}	2	{bfe}	2
{acf}	2	{b(f,g)}	3
{afe}	2	{bge}	2
{a(f,g)}	2	{cfe}	1
{age}	2	{(f,g)e}	2

3-len Seq	
{abe}	2
{abf}	2
{ace}	2
{acf}	2
{afe}	2
{a(f,g)}	2
{age}	2
{bfe}	2
{b(f,g)}	3
{bge}	2
{(f,g)e}	2

4-len Seq	
{abfe}	2
{ab(f,g)}	1
{a(f,g)e}	2
{b(f,g)e}	2

- {abf} {bfe} → {abfe}
- {abf} {bge}

1- length Seq	
{a}	3
{b}	4
{c}	2
{e}	3
{f}	4
{g}	3

2-len Seq	
{ab}	2
{ac}	2
{ae}	3
{af}	3
{ag}	2
{(a,b)}	2
{be}	3
{bf}	4
{bg}	3
{ce}	2
{cf}	2
{fe}	2
{(f,g)}	3
{ge}	2

3-len Seq	
{abe}	2
{abf}	2
{ace}	2
{acf}	2
{afe}	2
{a(f,g)}	2
{age}	2
{bfe}	2
{b(f,g)}	3
{bge}	2
{(f,g)e}	2

4-len Seq	
{abfe}	2
{a(f,g)e}	2
{b(f,g)e}	2


THANKS FOR WATCHING
I HOPE YOU HAVE LEARNED SOMETHING...