Report 1
Jia Yi ID:5133709256

In this machine problem, I implemented GSP and Spade algorithm for finding frequent subsequence.

The following are tests of time for running sequences with each length less than or equal to 30 and each length less than or equal to 60. The runtime only considers runtime for function GSP and Spade and excludes time for reading files and writing outputs.

Table 1--- Sequence length<=30

Sequence num	Minsup	Gsp(s)	Spade(s)
50	50	0.0005	0.0125
	25	8.0609	6.7909
100	100	0.0005	0.0180
	50	15.1900	17.7506
200	200	0.0006	0.0199
	100	15.9998	34.6596
1000	1000	0.0018	0.0594
	500	31.9995	211.5930

From This table, I conclude that GSP is faster than Spade at sequence number larger than 100. When sequence number increases, more time is needed to compute support for subsequence.

Table 2 ----Sequence length <=60

Sequence num	Minsup	GSP	Spade
50	50	0.0873	1.6869
	48	16.2959	26.7418
100	100	0.0488	1.5340
	98	2.7578	20.5132
200	200	0.0582	2.0395

	198	1.3521	22.2783
1000	1000	0.1611	6.9491
	998	0.4484	18.1136

From this table, I find that when sequence length is larger (60 is this table), runtime is smaller for large sequence number. The possible reason is that it is more difficult to find common subsequences with large support when sequence number increases.

Table 3 Runtime using sequencedb.txt

Sequence Num	Minsup	GSP(s)	Spade(s)
1062	1060	23.9954	850

Conclusion

GSP algorithm is faster than Spade when sequence length and number increases. Spade may be better for small sequence length and sequence number. Spade runtime increases much more than GSP runtime when sequence length increases.