1.a

Min(S)>= v function is an **anti-monotonicity** and **succinct** constraint.

Set:
S :

Item	Profit	Price
A	-30	30
B	0	5
C	50	25
D	20	20
E	-10	50

v = 25

For min(price)>=25, B and D fail, we can exclude all subsets of B or D.

A, C, E meet the requirement, all sets in S meet the requirement must be a subset belong to them.

1.b

Max(S)<= v function is an **anti-monotonicity** and **succinct** constraint.

Set:
S =

Item	Profit	Price
A	-30	30
B	0	5
C	50	25
D	20	20
E	-10	50

v = 25

For max(price)<=25, A and E fail, we can exclude all subsets of A or E.
B, C, D meet the requirement, all sets in S meet the requirement must be a subset belong to them.

1.c

T

1.d

Anti-monotonic will be more effective if few sets cannot meet the requirement (fail). Since anti-monotonic is fail once fail all, when there are few sets to be pruned, it would be a better solution.

Monotonic will be more effective if few sets can meet the requirement (succeed). Since monotonic is succeed once succeed forever, we can easily find the sets contain element meet requirement and get the right answer.

2.a

Since the minsup for d and e <3, the b's projected database should be:

(_c)(_)f
(_)c(f_)
(_c)f (bc)(f)

2.b

bb	1
bc	3
(bc)	2
bcf	3
bf	3

The frequent pattern should be <bc>, <bcf>and <bf>.

2.c

No candidate sequence needs to be generated and projected (parititioned) databases keep shrinking.