Q1. (6 pts in total)

Yes. The Apriori Algorithm can be adopted. (1 pt) The idea is similar to the original Apriori Algorithm discussed in class. The difference is that we use the profit of an itemset here instead of the total number of rows containing the itemset in the dataset. The reason why we can follow the framework of the Apriori Algorithm is because of the following property: if an itemset s has profit $\geq$50, then any proper subset of s has profit $\geq$50. (Or if an itemset s does not have profit $\geq$50, then any proper superset of s must not have profit $\geq$50.) (1 pt)

We can adopt the Apriori Algorithm as follows (2 pts):

1. $L_1 \leftarrow$ All items with profit $\geq$50
2. $k \leftarrow$2
3. While $L_{k-1} \neq \emptyset$ do
   - $C_k \leftarrow$ Generate candidates from L_{k-1} by the Join step and Prune step
   - Perform a counting step on C_k to obtain L_k
4. Return $\bigcup_i L_i$

Now we illustrate with the given example. First we compute L_1

	Frequency	Profit
A	0+3+0+1+6=10	10*10=100
B	0+4+0+0+0=4	4*10=40
C	3+0+1+3+0=7	7*10=70
D	2+0+3+5+0=10	10*10=100

So, $L_1 = \{A, C, D\}$. With Join & Prune, we can get $C_2 = \{AC, AD, CD\}$. Now we compute their profits

	Frequency	Profit
AC	0+0+0+1+0=1	1*10=10
AD	0+0+0+1+0=1	1*10=10
CD	2+0+1+3+0=6	6*10=60

So, $L_2 = \{CD\}$. The final output would be $\{A, C, D, CD\}$ (2 pts)

Q2. (4 pts in total)

No. The Apriori Algorithm cannot be adapted. (1 pt) In this problem, the following Apriori property cannot be satisfied: If an itemset s has profit $\geq$50, then any proper subset of s has profit $\geq$50. (Or if an itemset s does not have profit $\geq$50, then any proper superset of s must not have profit $\geq$50.) (1 pt)

The following shows an example that this property cannot be satisfied.

However, we can use another algorithm to compute the desired itemsets (2 pts)

	Frequency	Profit
CD	2+0+1+3+0=6	6*(6+4)=60
C	3+0+1+3+0=7	7*6=42

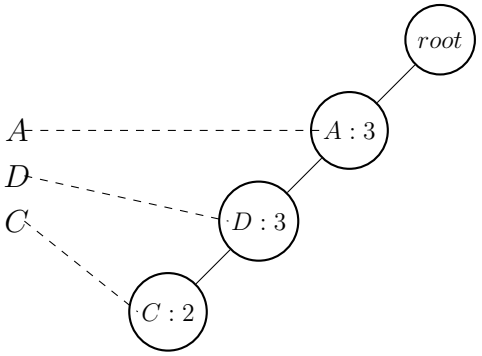
1. $O \leftarrow \emptyset$
2. For each itemset s with frequency ≥ 1
   - Find the profit of s
   - If the profit of s is greater than 50, $O \leftarrow O \cup \{s\}$
3. Return O

Q3. (7 pts in total)

Step 1: Scan all transactions and get the support count of each item.

Item	Support
A	3
D	3
C	2
B	1

Step 2: Scan all transactions and build the FP-tree. Note that item B will not be considered because its support does not satisfy the threshold.



Step 3: Build conditional FP-tree for each item.

Item	Conditional FP-tree	Frequent itemsets
$C:2$	$root \rightarrow (A:2) \rightarrow (D:2)$	$\{C:2, CD:2, AC:2, AD:2, ACD:2\}$
$D:3$	$root \rightarrow (A:3)$	$\{D:3, AD:3\}$
$A:3$	$root$	$\{A:3\}$

Step 4: Generate all frequent itemsets $\{A, C, D, AC, AD, CD, ACD\}$(as shown in the third column of the above table)

Suggested Marking Scheme for Q3:

- 1 pt for count the support of items in step 1;

- 2 pts points for constructing the correct FP-tree in step 2;
- 2 pts points for correct conditional FP-tree in step 3;
- 2 pts points for the correct final result.

Q4. (3 pts in total, COMP7118 Only)

No, the FP-tree structure obtained remains the same even if the transactions are shuffled (1 pt). We can prove this by contradiction. Assume the tree structure is different, it implies that at least one transaction will be have a different encoding in the prefix tree, which concludes that the transaction will be reconstructed differently. However, the transaction itself remains unique and unchanged. This contradicts the assumption. (2 pts, any reasonable argument should be accepted)

Q5. (3 pts in total, COMP8118 Only)

An itemset is *maximal frequent* if none of its immediate proper superset is frequent. In the fourth step of FP-tree growth algorithm, for each item i, we can identify the set of maximal itemsets that contain i from i's conditional FP-tree, denoted by $S(i)$. If an itemset $I \in S(i)$ is a maximal frequent itemset, it must not be a proper subset of any element from $S(j)$, where j is another item. To make it work, we need compute $S(i)$ for every item, then start from the candidate itemsets of the largest size, perform an iterative scan until no smaller sized itemsets can be pruned. In this way, it will find us all the maximal frequent itemsets.

An itemset is closed frequent if none of its immediate proper superset share the same frequency. We can make use of the counter associated with each item's conditional FP-tree to directly output the answer. The rule is, any prefix path is a closed frequent itemset iff adding one more item to the path leads to a smaller count.

Suggested Marking Scheme for Q5:

- If the maximal/closed frequent itemset concept is explained, at least 1 pt should be granted
- If the explanation makes sense for one of two cases, at least 2 pts should be granted. Full marks should be granted if both cases are well explained.