

Data Structure Lab. Project #2

2020년 10월 7일

Due date: 2020년 11월 6일 금요일 23:59:59 까지

본 프로젝트에서는 FP-Growth와 B+-Tree를 이용하여 상품 추천 프로그램을 구현한다. 이 프로그램은 장바구니 데이터에서 같이 구매한 상품들을 받아 FP-Growth를 구축한다. FP-Growth는 상품들의 연관성을 Tree 구조로 저장하고 있는 FP-Tree와 상품별 빈도수 및 정보, 해당 상품과 연결된 FP-Tree의 상품 노드들을 관리하는 Header Table로 구성된다. FP-Growth 구축 단계에서 연관된 상품들을 묶은 Frequent Pattern의 결과는 result.txt에 빈도수, 상품 순으로 저장한다. Frequent Pattern들이 저장된 result.txt는 BTLOAD 명령어를 통해 빈도수를 기준으로 B+-Tree에 저장된다. B+-Tree는 IndexNode와 DataNode로 구성된다. IndexNode는 DataNode를 찾기 위한 Node이고 DataNode는 해당 빈도수를 가지는 Frequent Pattern들이 저장된 Node이다. 채점 기준은 FP-Tree와 Header Table 생성 및 연결과 B+-Tree이며, FP-Growth의 결과인 result.txt(Frequent Pattern)는 가산점으로 부여된다.

자료구조의 구축 방법과 조건에 대한 자세한 설명은 program implementation에서 설명한다.

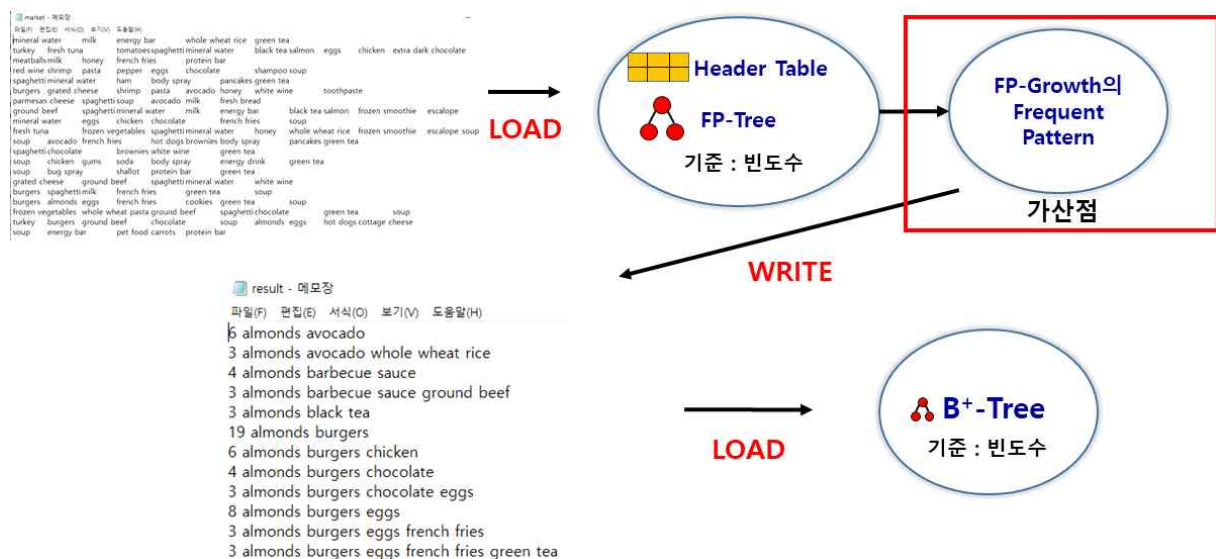


그림 1. 상품 관리 프로그램 구조

□ Program implementation

1) FP-Growth

- 주어진 market.txt에 저장된 데이터를 이용하여 구축한다. market.txt에서 같이 구매한 물품은 줄 단위로 구분되어 있다.
- 프로그램 구현 시 주어진 FPNODE 클래스를 통해 FP-Tree를 구현하고 주어진 Header Table 클래스를 통해 Header Table을 구현한다.

- 상품 정보는 항상 고유하며, 소문자로 표기한다고 가정한다.
- FP-Growth의 Threshold 값은 고정되어 있지 않으며 멤버 변수로 변경할 수 있다.
 - Threshold 값은 2 이상으로 설정한다고 가정한다.
- Header Table, FP-Tree, B+-Tree의 구축 방법과 조건에 대한 자세한 설명은 아래에서 추가 설명한다.
- FP-Growth가 생성되면 그 결과를 result.txt에 저장한다. result.txt 저장을 구현할 시 가산점이 부여된다.

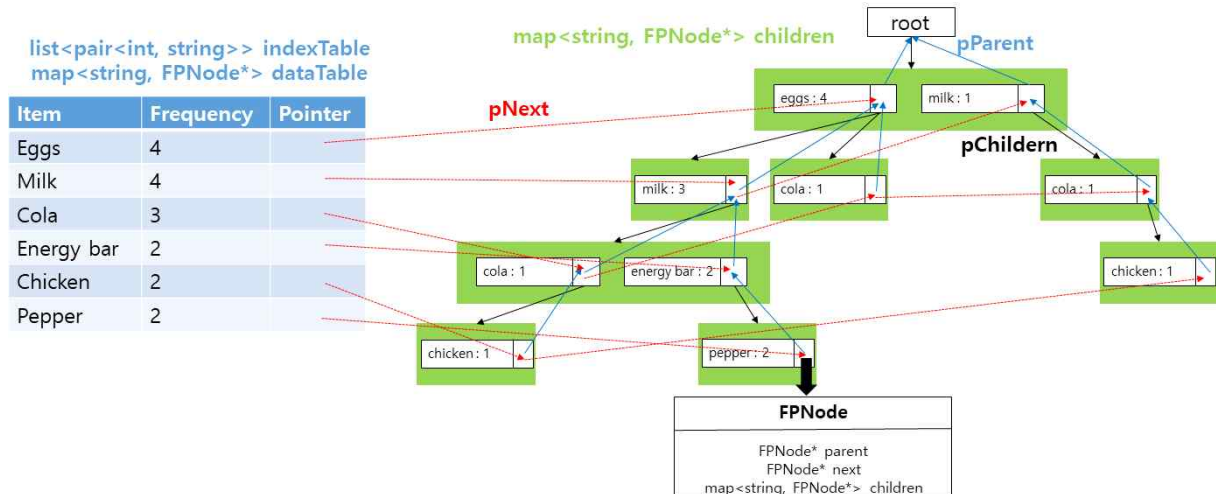


그림 2. FP-Growth의 예

2) Header Table

- Header Table은 인덱스와 데이터로 구성한다.
- threshold보다 작은 상품들도 저장한다.
- 인덱스는 빈도수를 기준으로 상품들이 정렬된 변수이다. 이 변수는 stl의 list를 사용하며 list 안에는 pair를 통해 빈도수를 key로 상품을 value로 저장한다.
 - list<pair<int, string>> indexTable
- 데이터는 상품과 상품에 연결되는 노드를 가지고 있는 변수이다. 이 변수는 stl의 map 컨테이너 형태로 상품을 key로 하여 상품과 연결되는 FP-Tree의 Node를 저장한다.
 - map<string, FPNode*> dataTable
- 인덱스에서는 빈도수를 기준으로 오름차순과 내림차순으로 정렬을 할 수 있는 함수가 구현되어야 한다. 정렬은 list에서 제공되는 sort 함수를 이용하여 구현한다.

3) FP-Tree

- FP-Tree 클래스를 따로 생성하지 않고 FPNode를 이용하여 구축한다.
- root에서 자식 노드를 제외한 변수들은 NULL 값을 갖는다.
- 자식 노드들은 map 컨테이너 형태로 저장하며, 부모 노드를 가리키는 노드가 존재한다. key의 string은 상품명을 저장하고 value의 FPNode*는 해당 상품의 빈도수 정보 및 연결된 Node 정보를 저장한다.
 - map<string, FPNode*> children
- FP-Tree에 저장된 노드들은 Header Table에서 같은 상품 노드들끼리 연결되어야 한다.

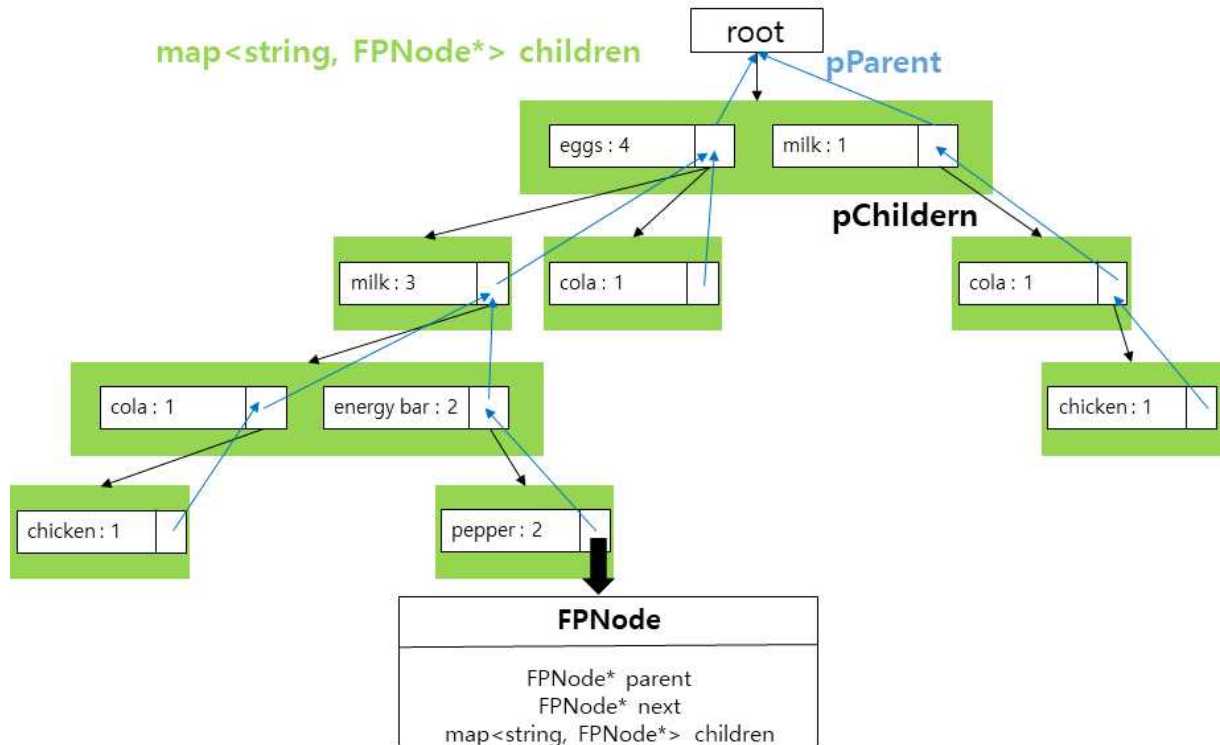


그림 3. FP-Tree의 예

- FP-Tree에서 각 연결된 연관 상품에 따라 빈도수를 정확히 설정해야 한다.
- FP-Tree에 연결되는 상품 순서는 빈도수를 기준으로 내림차순으로 결정이 되며, Header Table의 정렬 기준에 따라 결정이 된다.
- 빈도수가 같은 경우 Header Table에서 indexTable의 정렬 기준에 따라 결정이 된다.
- 저장되는 데이터는 상품명과 상품 노드이며 상품 노드에는 빈도수 정보, 부모 노드 정보, 자식 노드 정보가 저장되어야 한다.

4) B+-Tree

- result.txt에 저장된 데이터를 이용하여 구축한다.
- result.txt는 빈도수가 제일 처음으로 주어지며 그 뒤에 “\t”을 구분자로 하여 연관된 상품들을 저장한 파일이다.
- B+-Tree는 그림 3과 같이 빈도수를 기준으로 정렬된다.
- B+-Tree는 인덱스 노드(BpTreeIndexNode)와 데이터 노드 (BpTreeDataNode)로 구성되며, 각 노드 클래스는 B⁺-tree 노드 클래스(BpTreeNode)를 상속받는다.
- 데이터 노드는 단말 노드로, 해당 빈도수에 속하는 Frequent Pattern이 저장된 FrequentPatternNode를 map 컨테이너 형태로 가지고 있으며 가장 왼쪽 자식을 가리키는 포인터를 따로 가지고 있다.
- B+-Tree의 차수 ORDER(m)는 고정되어 있지 않으며 멤버 변수로 변경할 수 있다.

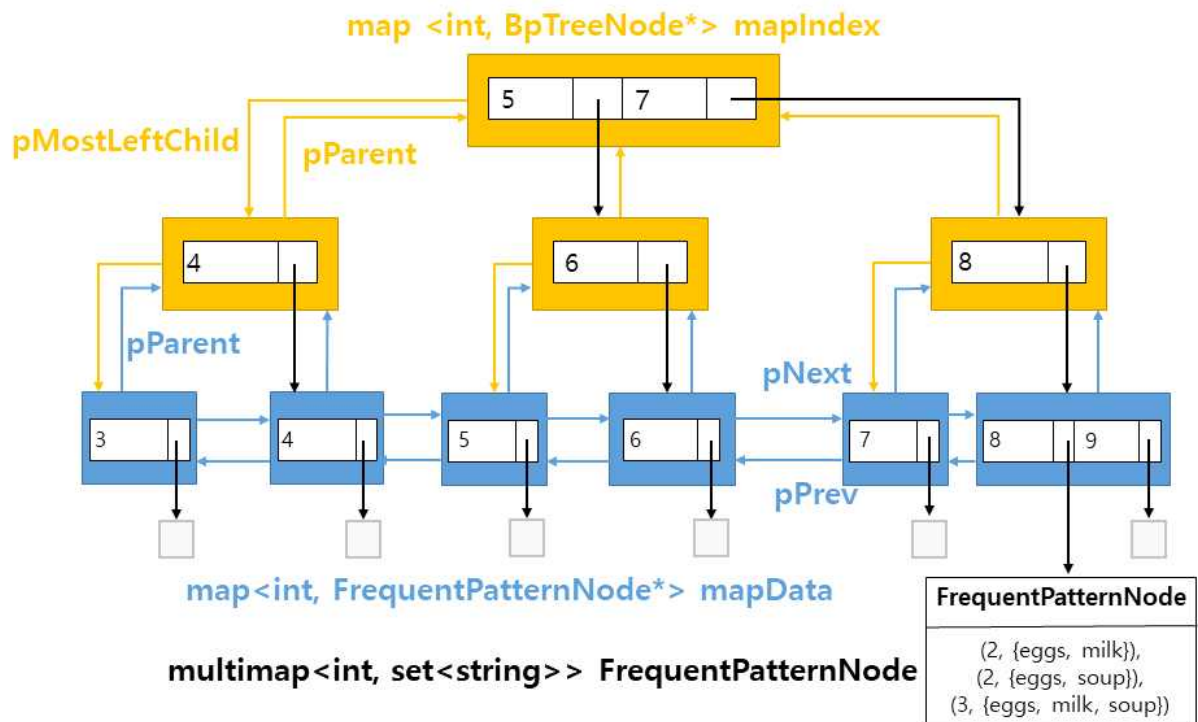


그림 3. B+-Tree의 예

5) FrequentPatternNode

- FrequentPatternNode는 Frequent Pattern 정보를 multimap 컨테이너 형태로 가지고 있다.
 - `multimap<int, set<string>> FrequentPatternList`
- 각 mapData의 빈도수에 해당하는 Frequent Pattern의 정보를 저장하고 있어야 하며 key로 Frequent Pattern의 크기를 저장하고 value로 Frequent Pattern의 원소를 저장해야 한다.
- value의 Frequent Pattern의 원소는 stl의 set 컨테이너 형태로 가지고 있다. set에는 원소에 해당하는 상품명이 저장돼야 한다.
 - `set<string> item`
- Frequent Pattern 중 공집합 이거나 원소가 하나인 집합은 저장하지 않는다.

□ Functional Requirements

- 출력 포맷은 포맷에 대한 예시일 뿐 실제 출력되는 데이터들과는 차이가 있을 수 있습니다.

표 1. 명령어 사용 예 및 기능 설명

명령어	명령어 사용 예 및 기능
LOAD	<p>사용 예) LOAD</p> <p>텍스트 파일의 데이터 정보를 불러오는 명령어로, 텍스트 파일에 데이터 정보가 존재할 경우 텍스트 파일을 읽어 FP-Growth를 생성한다. 만약 텍스트 파일이 존재하지 않거나 자료구조에 이미 데이터가 들어가 있으면 알맞은 예러 코드를 출력한다. 사용되는 텍스트 파일의 이름은</p>

	<p>아래와 같으며 파일 이름을 수정하지 않는다.</p> <p>텍스트 파일: market.txt</p> <p>*데이터 조건</p> <ul style="list-style-type: none"> - 상품명은 무조건 소문자로 주어지며 그 이외의 입력은 없다고 가정한다. - “\t”를 구분자로 하여 상품을 구분한다. - 같이 구매한 물품들은 줄 단위로 구분된다. <p>출력 포맷 예시)</p> <pre>=====LOAD===== Success ===== =====LOAD===== ===== ERROR 100 ===== =====</pre>
BTLOAD	<p>사용 예) BTLOAD</p> <p>텍스트 파일의 데이터 정보를 불러오는 명령어로, 텍스트 파일에 데이터 정보가 존재할 경우 텍스트 파일을 읽어 B+-Tree에 저장한다. 만약 텍스트 파일이 존재하지 않거나 자료구조에 이미 데이터가 들어가 있으면 알맞은 에러 코드를 출력한다. 사용되는 텍스트 파일의 이름은 아래와 같으며 파일 이름을 수정하지 않는다.</p> <p>텍스트 파일: result.txt</p> <p>*데이터 조건</p> <ul style="list-style-type: none"> - 첫 번째 값은 빈도수이고 그 이후 값은 상품 정보이다. - “\t”를 구분자로 하여 상품을 구분한다. - Frequent Pattern에 속한 상품들은 줄 단위로 구분된다. <p>출력 포맷 예시)</p> <pre>=====BTLOAD===== Success ===== =====BTLOAD===== ===== ERROR 200 ===== =====</pre>

PRINT_ITEMLIST	<p>사용 예) PRINT_ITEMLIST</p> <p>FP-Growth의 Header Table에 저장된 상품들을 내림차순으로 출력하는 명령어로 threshold보다 작은 빈도수를 가진 상품도 출력한다. Header Table이 비어 있는 경우 에러 코드를 출력한다.</p> <p>출력 포맷 예시)</p> <pre> =====PRINT_ITEMLIST===== Item Frequency soup 12 spaghetti 9 green tea 9 mineral water 7 milk 5 ... ===== =====PRINT_ITEMLIST===== ===== ERROR 300 ===== ===== </pre>
PRINT_FPTREE	<p>사용 예) PRINT_FPTREE</p> <p>FP-Growth의 FP-Tree 정보를 출력하는 명령어로 아래 출력 기준에 맞춰 출력해주는 명령어이다. Header Table을 먼저 오름차순으로 정렬하고 threshold 이상의 상품들을 출력한다. FP-Tree가 비어 있는 경우 에러 코드를 출력한다.</p> <p>*출력 조건</p> <ul style="list-style-type: none"> - Header Table의 오름차순 순으로 FP-Tree의 path를 출력한다. - threshold보다 작은 상품은 넘어간다. - Header Table의 상품을 {상품명,빈도수}로 출력한다. - 해당 상품과 연결된 FP-Tree의 path들을 (상품명,빈도수)로 root 노드 전까지 연결된 부모 노드를 출력한다. - 해당 상품과 연결된 다음 노드들이 없을 때까지 출력하며 다음 노드로 이동하면 다음 줄로 이동한다. <p>출력 포맷 예시)</p> <pre> =====PRINT_FPTREE===== {StandardItem,Frequency} (Path_Item,Frequency) {almonds,2} (almonds,1) (burgers,1) (eggs,1) (french fries,2) (green tea,4) </pre>

	<pre> (soup,12) (almonds,1) (hot dogs,1) (turkey,1) (burgers,1) (ground beef,1) (chocolate,2) (eggs,2) (soup,12) {black tea,2} (black tea,1) (fresh tuna,1) (salmon,1) (turkey,1) (chicken,1) (eggs,1) (mineral water,3) (spaghetti,5) (black tea,1) (escalope,1) (frozen smoothie,1) (salmon,1) (energy bar,1) (ground beef,1) (milk,1) (mineral water,3) (spaghetti,5) {brownies,2} (brownies,1) (hot dogs,1) (pancakes,1) (avocado,1) (body spray,1) (french fries,2) (green tea,4) (soup,12) (brownies,1) (white wine,1) (chocolate,1) (green tea,2) (spaghetti,5) ... ===== =====PRINT_FPTREE===== ===== ERROR 400 ===== ===== </pre>
PRINT_MIN	<p>사용 예) PRINT_MIN soup 2</p> <p>B+-Tree에 저장된 Frequent Pattern 중 입력된 상품과 두 번째 인자로 받은 최소 빈도수 이상의 값을 가지는 Frequent Pattern을 출력하는 명령어이다. 첫 번째 인자로 상품명을 입력받고 두 번째 인자로 최소 빈도수를 받는다.</p> <p>명령이 실행되면 입력받은 최소 빈도수를 기준으로 B+-Tree에서 탐색한다. 탐색이 끝나면 B+-Tree에 저장된 Frequent Pattern 중 최소 빈도수 이상을 가지는 Frequent Pattern을 출력하며 이동한다.</p> <p>출력할 Frequent Pattern이 없거나 B+-Tree가 비어 있는 경우 에러 코드를 출력한다.</p> <p>출력 포맷 예시)</p> <pre> =====PRINT_MIN===== StandardItem FrequentPatternFrequency soup -> {almonds} 2 soup -> {avocado} 2 soup -> {body spray} 2 soup -> {chicken} 2 soup -> {frozen vegetables} 2 soup -> {ground beef} 2 soup -> {hot dogs} 2 </pre>

	<p>soup -> {milk} 2</p> <p>...</p> <p>=====</p> <p>=====PRINT_MIN=====</p> <p>===== ERROR 500 =====</p> <p>=====</p>															
PRINT_CONFIDENCE	<p>사용 예) PRINT_CONFIDENCE soup 0.3</p> <p>B+-Tree에 저장된 Frequent Pattern 중 입력된 상품과 두 번째 인자로 받은 연관율 이상의 값을 가지는 Frequent Pattern을 출력하는 명령어이다. 첫 번째 인자로 상품명을 입력받고 두 번째 인자로 연관율을 받는다.</p> <p>연관율은 $\text{연관율} = \frac{\text{부분집합의 빈도수}}{\text{해당 상품의 총 빈도수}}$로 계산이 된다.</p> <p>명령이 실행되면 입력받은 연관율과 해당 상품의 총 빈도수의 곱보다 큰 빈도수를 B+-Tree에서 탐색한다. 탐색이 끝나면 B+-Tree에 저장된 연관율 이상의 Frequent Pattern을 출력하며 이동한다.</p> <p>출력할 Frequent Pattern이 없거나 B+-Tree가 비어 있는 경우 에러 코드를 출력한다.</p> <p>출력 포맷 예시)</p> <p>=====PRINT_CONFIDENCE=====</p> <table><tr><td>StandardItem</td><td>FrequentPatternFrequency</td><td>Confidence</td></tr><tr><td>soup -> {chocolate}</td><td>4</td><td>0.33</td></tr><tr><td>soup -> {eggs}</td><td>4</td><td>0.33</td></tr><tr><td>soup -> {french fries}</td><td>4</td><td>0.33</td></tr><tr><td>...</td><td></td><td></td></tr></table> <p>=====</p> <p>=====PRINT_CONFIDENCE=====</p> <p>===== ERROR 600 =====</p> <p>=====</p>	StandardItem	FrequentPatternFrequency	Confidence	soup -> {chocolate}	4	0.33	soup -> {eggs}	4	0.33	soup -> {french fries}	4	0.33	...		
StandardItem	FrequentPatternFrequency	Confidence														
soup -> {chocolate}	4	0.33														
soup -> {eggs}	4	0.33														
soup -> {french fries}	4	0.33														
...																
PRINT_RANGE	<p>사용 예) PRINT_RANGE soup 3 4</p> <p>B+-Tree에 저장된 Frequent Pattern을 출력하는 명령어로, 첫 번째 인자로 상품명을 입력하고 두 번째 인자로 최소 빈도수, 세 번째 인자로 최대 빈도수를 입력받는다. 3개의 인자가 모두 입력되지 않거나 형식이 다르면 에러 코드를 출력한다.</p> <p>명령이 실행되면 최소 빈도수를 가지고 B+-Tree에서 Frequent Pattern을 탐색한다. 탐색이 끝나면 최대 빈도수까지 B+-Tree에 저장</p>															

	<p>된 Frequent Pattern을 출력하며 이동한다. 출력할 Frequent Pattern이 없거나 B+-Tree가 비어 있는 경우 에러 코드를 출력한다.</p> <p>출력 포맷 예시)</p> <pre> =====PRINT_RANGE===== StandardItem FrequentPatternFrequency soup -> {burgers} 3 soup -> {chocolate, eggs} 3 soup -> {french fries, green tea} 3 soup -> {chocolate} 4 ... ===== =====PRINT_RANGE===== ===== ERROR 700 ===== ===== </pre>
SAVE	<p>사용 예) SAVE</p> <p>FP-Growth의 상품들의 연관성 결과를 저장하는 명령어로 생성된 Frequent Pattern들을 result.txt에 저장한다. 저장 포맷은 ‘\t’를 구분자로 하여 빈도수, 상품명 순으로 저장한다. FP Growth의 frequentPattern이 비어 있는 경우 예외처리를 한다.</p> <p>출력 포맷 예시)</p> <pre> =====SAVE===== Success ===== =====SAVE===== ===== ERROR 800 ===== ===== result.txt 결과 예시) 2 almonds burgers 2 almonds burgers eggs 2 almonds burgers eggs soup 2 almonds burgers soup 2 almonds eggs 2 almonds eggs soup 2 almonds soup 2 avocado soup 2 black tea mineral water 2 black tea mineral water salmon </pre>

	2 black tea mineral water salmon spaghetti 2 black tea mineral water spaghetti 2 black tea salmon 2 black tea salmon spaghetti 2 black tea spaghetti 3 body spray green tea 2 body spray green tea pancakes ...
EXIT	<p>사용 예) EXIT</p> <p>프로그램상의 메모리를 해제하며, 프로그램을 종료한다.</p> <p>출력 포맷 예시)</p> <pre>=====EXIT===== Success =====</pre>

□ Requirements in implementation

- ✓ 모든 명령어는 command.txt에 저장하여 순차적으로 읽고 처리한다.
- ✓ 모든 명령어는 반드시 대문자로 입력한다.
- ✓ 명령어에 인자(Parameter)가 모자라거나 필요 이상으로 입력받으면 에러 코드를 출력한다.
- ✓ 예외처리에 대해 반드시 에러 코드를 출력한다.
- ✓ 출력은 “출력 포맷”을 반드시 따라 한다.
- ✓ log.txt 파일에 출력 결과를 반드시 저장한다.
 - log.txt가 이미 존재할 경우 텍스트 파일 가장 뒤에 이어서 추가로 저장한다.
- ✓ 읽어야 할 텍스트 파일이 존재하지 않으면 해당 텍스트 파일을 생성한 뒤 진행하도록 한다.

□ 동작 별 에러 코드

동작	에러 코드
LOAD	100
BTLOAD	200
PRINT_ITEMLIST	300
PRINT_FPTREE	400
PRINT_MIN	500
PRINT_CONFIDENCE	600
PRINT_RANGE	700
SAVE	800

□ 동작 예시

market.txt (threshold = 2)									
mineral water	milk	energy bar	whole wheat rice	green tea					
turkey	fresh tuna	tomatoes	spaghetti	mineral water	black tea	salmon	eggs	chicken	extra dark chocolate
meatballs	milk	honey	french fries	protein bar					
red wine	shrimp	pasta	pepper	eggs	chocolate		shampoo	soup	
spaghetti	mineral water	ham	body spray	pancakes	green tea				
burgers	grated cheese	shrimp	pasta	avocado	honey	white wine		toothpaste	
parmesan	cheese	spaghetti	soup	avocado	milk	fresh bread			
ground beef	spaghetti	mineral water	milk	energy bar		black tea	salmon	frozen smoothie	escalope
mineral water	eggs	chicken	chocolate	french fries	soup				
fresh tuna	frozen vegetables	spaghetti	mineral water	honey	whole wheat rice	frozen smoothie	escalope	soup	
soup	avocado	french fries	hot dogs	brownies	body spray	pancakes	green tea		
spaghetti	chocolate		brownies	white wine	green tea				
soup	chicken	gums	soda	body spray	energy drink	green tea			
soup	bug spray		shallot	protein bar	green tea				
grated cheese	ground beef	spaghetti	mineral water	white wine					
burgers	spaghetti	milk	french fries	green tea	soup				
burgers	almonds	eggs	french fries	cookies	green tea	soup			
frozen vegetables	whole wheat	pasta	ground beef	spaghetti	chocolate	green tea	soup		
turkey	burgers	ground beef	chocolate	soup	almonds	eggs	hot dogs	cottage cheese	
soup	energy bar	pet food	carrots	protein bar					
command.txt									
LOAD									
LOAD									
PRINT_ITEMLIST									
PRINT_FPTREE									
SAVE									
BTLOAD									
BTLOAD									
PRINT_MIN soup 2									
PRINT_CONFIDENCE soup 0.3									
PRINT_RANGE soup 3 5									
result.txt									
2	almonds	burgers							
2	almonds	burgers	eggs						
2	almonds	burgers	eggs	soup					
2	almonds	burgers	soup						
2	almonds	eggs							
2	almonds	eggs	soup						
2	almonds	soup							
2	avocado	soup							
2	black tea	mineral water							
2	black tea	mineral water	salmon						
2	black tea	mineral water	salmon	spaghetti					
2	black tea	mineral water	spaghetti						
2	black tea	salmon							
2	black tea	salmon	spaghetti						
2	black tea	spaghetti							
3	body spray	green tea							
2	body spray	green tea	pancakes						
2	body spray	green tea	soup						
2	body spray	pancakes							
2	body spray	soup							

2	brownies	green tea		
2	burgers	eggs		
2	burgers	eggs	soup	
2	burgers	french fries		
2	burgers	french fries	green tea	
2	burgers	french fries	green tea soup	
2	burgers	french fries	soup	
2	burgers	green tea		
2	burgers	green tea soup		
3	burgers	soup		
2	chicken	eggs		
2	chicken	eggs	mineral water	
2	chicken	mineral water		
2	chicken	soup		
3	chocolate	eggs		
3	chocolate	eggs	soup	
2	chocolate	green tea		
2	chocolate	green tea spaghetti		
2	chocolate	ground beef		
2	chocolate	ground beef	soup	
4	chocolate	soup		
2	chocolate	spaghetti		
2	eggs	french fries		
2	eggs	french fries	soup	
2	eggs	mineral water		
4	eggs	soup		
2	eggs	turkey		
2	energy bar	milk		
2	energy bar	milk	mineral water	
2	energy bar	mineral water		
2	escalope	frozen smoothie		
2	escalope	frozen smoothie	mineral water	
2	escalope	frozen smoothie	mineral water	spaghetti
2	escalope	frozen smoothie	spaghetti	
2	escalope	mineral water		
2	escalope	mineral water	spaghetti	
2	escalope	spaghetti		
3	french fries	green tea		
3	french fries	green tea soup		
2	french fries	milk		
4	french fries	soup		
2	fresh tuna	mineral water		
2	fresh tuna	mineral water	spaghetti	
2	fresh tuna	spaghetti		
2	frozen smoothie	mineral water		
2	frozen smoothie	mineral water	spaghetti	
2	frozen smoothie	spaghetti		
2	frozen vegetables	soup		
2	frozen vegetables	soup	spaghetti	
2	frozen vegetables	spaghetti		
2	grated cheese	white wine		
2	green tea	milk		
2	green tea	mineral water		

2	green tea	pancakes		
6	green tea	soup		
2	green tea	soup	spaghetti	
4	green tea	spaghetti		
2	ground beef	mineral water		
2	ground beef	mineral water	spaghetti	
2	ground beef	soup		
3	ground beef	spaghetti		
2	hot dogs	soup		
2	milk	mineral water		
2	milk	soup		
2	milk	soup	spaghetti	
3	milk	spaghetti		
2	mineral water	salmon		
2	mineral water	salmon	spaghetti	
2	mineral water	soup		
5	mineral water	spaghetti		
2	mineral water	whole wheat rice		
2	pasta	shrimp		
2	protein bar	soup		
2	salmon	spaghetti		
4	soup	spaghetti		
2	spaghetti	white wine		

log.txt

=====LOAD=====

Success

=====

=====LOAD=====

===== ERROR 100 =====

=====

=====PRINT_ITEMLIST=====

Item Frequency

soup 12

spaghetti 9

green tea 9

mineral water 7

milk 5

french fries 5

eggs 5

chocolate 5

ground beef 4

burgers 4

white wine 3

protein bar 3

honey 3

energy bar 3

chicken 3

body spray 3

avocado 3

```

whole wheat rice 2
turkey 2
shrimp 2
salmon 2
pasta 2
pancakes 2
hot dogs 2
grated cheese 2
frozen vegetables 2
frozen smoothie 2
fresh tuna 2
escalope 2
brownies 2
black tea 2
almonds 2
whole wheat pasta 1
toothpaste 1
tomatoes 1
soda 1
shampoo 1
shallot 1
red wine 1
pet food 1
pepper 1
parmesan cheese 1
meatballs 1
ham 1
gums 1
fresh bread 1
extra dark chocolate 1
energy drink 1
cottage cheese 1
cookies 1
carrots 1
bug spray 1
=====

=====PRINT_FPTREE=====
{StandardItem,Frequency} (Path_Item,Frequency)
{almonds,2}
(almonds,1) (burgers,1) (eggs,1) (french fries,2) (green tea,4) (soup,12)
(almonds,1) (hot dogs,1) (turkey,1) (burgers,1) (ground beef,1) (chocolate,2) (eggs,2) (soup,12)
{black tea,2}
(black tea,1) (fresh tuna,1) (salmon,1) (turkey,1) (chicken,1) (eggs,1) (mineral water,3) (spaghetti,5)
(black tea,1) (escalope,1) (frozen smoothie,1) (salmon,1) (energy bar,1) (ground beef,1) (milk,1) (mineral
water,3) (spaghetti,5)
{brownies,2}
(brownies,1) (hot dogs,1) (pancakes,1) (avocado,1) (body spray,1) (french fries,2) (green tea,4) (soup,12)
(brownies,1) (white wine,1) (chocolate,1) (green tea,2) (spaghetti,5)
{escalope,2}
(escalope,1) (frozen smoothie,1) (salmon,1) (energy bar,1) (ground beef,1) (milk,1) (mineral water,3)
(spaghetti,5)
(escalope,1) (fresh tuna,1) (frozen smoothie,1) (frozen vegetables,1) (whole wheat rice,1) (honey,1) (mineral

```

water,1) (spaghetti,4) (soup,12)
 {fresh tuna,2}
 (fresh tuna,1) (salmon,1) (turkey,1) (chicken,1) (eggs,1) (mineral water,3) (spaghetti,5)
 (fresh tuna,1) (frozen smoothie,1) (frozen vegetables,1) (whole wheat rice,1) (honey,1) (mineral water,1)
 (spaghetti,4) (soup,12)
 {frozen smoothie,2}
 (frozen smoothie,1) (salmon,1) (energy bar,1) (ground beef,1) (milk,1) (mineral water,3) (spaghetti,5)
 (frozen smoothie,1) (frozen vegetables,1) (whole wheat rice,1) (honey,1) (mineral water,1) (spaghetti,4)
 (soup,12)
 {frozen vegetables,2}
 (frozen vegetables,1) (whole wheat rice,1) (honey,1) (mineral water,1) (spaghetti,4) (soup,12)
 (frozen vegetables,1) (ground beef,1) (chocolate,1) (green tea,2) (spaghetti,4) (soup,12)
 {grated cheese,2}
 (grated cheese,1) (pasta,1) (shrimp,1) (avocado,1) (honey,1) (white wine,1) (burgers,1)
 (grated cheese,1) (white wine,1) (ground beef,1) (mineral water,3) (spaghetti,5)
 {hot dogs,2}
 (hot dogs,1) (pancakes,1) (avocado,1) (body spray,1) (french fries,2) (green tea,4) (soup,12)
 (hot dogs,1) (turkey,1) (burgers,1) (ground beef,1) (chocolate,2) (eggs,2) (soup,12)
 {pancakes,2}
 (pancakes,1) (body spray,1) (mineral water,1) (green tea,2) (spaghetti,5)
 (pancakes,1) (avocado,1) (body spray,1) (french fries,2) (green tea,4) (soup,12)
 {pasta,2}
 (pasta,1) (shrimp,1) (chocolate,2) (eggs,2) (soup,12)
 (pasta,1) (shrimp,1) (avocado,1) (honey,1) (white wine,1) (burgers,1)
 {salmon,2}
 (salmon,1) (turkey,1) (chicken,1) (eggs,1) (mineral water,3) (spaghetti,5)
 (salmon,1) (energy bar,1) (ground beef,1) (milk,1) (mineral water,3) (spaghetti,5)
 {shrimp,2}
 (shrimp,1) (chocolate,2) (eggs,2) (soup,12)
 (shrimp,1) (avocado,1) (honey,1) (white wine,1) (burgers,1)
 {turkey,2}
 (turkey,1) (chicken,1) (eggs,1) (mineral water,3) (spaghetti,5)
 (turkey,1) (burgers,1) (ground beef,1) (chocolate,2) (eggs,2) (soup,12)
 {whole wheat rice,2}
 (whole wheat rice,1) (energy bar,1) (milk,1) (mineral water,1) (green tea,1)
 (whole wheat rice,1) (honey,1) (mineral water,1) (spaghetti,4) (soup,12)
 {avocado,3}
 (avocado,1) (honey,1) (white wine,1) (burgers,1)
 (avocado,1) (milk,1) (spaghetti,4) (soup,12)
 (avocado,1) (body spray,1) (french fries,2) (green tea,4) (soup,12)
 {body spray,3}
 (body spray,1) (mineral water,1) (green tea,2) (spaghetti,5)
 (body spray,1) (french fries,2) (green tea,4) (soup,12)
 (body spray,1) (chicken,1) (green tea,4) (soup,12)
 {chicken,3}
 (chicken,1) (eggs,1) (mineral water,3) (spaghetti,5)
 (chicken,1) (chocolate,1) (eggs,1) (french fries,1) (mineral water,1) (soup,12)
 (chicken,1) (green tea,4) (soup,12)
 {energy bar,3}
 (energy bar,1) (milk,1) (mineral water,1) (green tea,1)
 (energy bar,1) (ground beef,1) (milk,1) (mineral water,3) (spaghetti,5)
 (energy bar,1) (protein bar,1) (soup,12)
 {honey,3}

(honey,1) (protein bar,1) (french fries,1) (milk,1)
 (honey,1) (white wine,1) (burgers,1)
 (honey,1) (mineral water,1) (spaghetti,4) (soup,12)
 {protein bar,3}
 (protein bar,1) (french fries,1) (milk,1)
 (protein bar,1) (green tea,4) (soup,12)
 (protein bar,1) (soup,12)
 {white wine,3}
 (white wine,1) (burgers,1)
 (white wine,1) (chocolate,1) (green tea,2) (spaghetti,5)
 (white wine,1) (ground beef,1) (mineral water,3) (spaghetti,5)
 {burgers,4}
 (burgers,1)
 (burgers,1) (french fries,1) (milk,1) (green tea,2) (spaghetti,4) (soup,12)
 (burgers,1) (eggs,1) (french fries,2) (green tea,4) (soup,12)
 (burgers,1) (ground beef,1) (chocolate,2) (eggs,2) (soup,12)
 {ground beef,4}
 (ground beef,1) (milk,1) (mineral water,3) (spaghetti,5)
 (ground beef,1) (mineral water,3) (spaghetti,5)
 (ground beef,1) (chocolate,1) (green tea,2) (spaghetti,4) (soup,12)
 (ground beef,1) (chocolate,2) (eggs,2) (soup,12)
 {chocolate,5}
 (chocolate,2) (eggs,2) (soup,12)
 (chocolate,1) (eggs,1) (french fries,1) (mineral water,1) (soup,12)
 (chocolate,1) (green tea,2) (spaghetti,5)
 (chocolate,1) (green tea,2) (spaghetti,4) (soup,12)
 {eggs,5}
 (eggs,1) (mineral water,3) (spaghetti,5)
 (eggs,2) (soup,12)
 (eggs,1) (french fries,1) (mineral water,1) (soup,12)
 (eggs,1) (french fries,2) (green tea,4) (soup,12)
 {french fries,5}
 (french fries,1) (milk,1)
 (french fries,1) (mineral water,1) (soup,12)
 (french fries,2) (green tea,4) (soup,12)
 (french fries,1) (milk,1) (green tea,2) (spaghetti,4) (soup,12)
 {milk,5}
 (milk,1) (mineral water,1) (green tea,1)
 (milk,1)
 (milk,1) (spaghetti,4) (soup,12)
 (milk,1) (mineral water,3) (spaghetti,5)
 (milk,1) (green tea,2) (spaghetti,4) (soup,12)
 {mineral water,7}
 (mineral water,1) (green tea,1)
 (mineral water,3) (spaghetti,5)
 (mineral water,1) (green tea,2) (spaghetti,5)
 (mineral water,1) (soup,12)
 (mineral water,1) (spaghetti,4) (soup,12)
 {green tea,9}
 (green tea,1)
 (green tea,2) (spaghetti,5)
 (green tea,4) (soup,12)
 (green tea,2) (spaghetti,4) (soup,12)


```

{spaghetti,9}
(spaghetti,5)
(spaghetti,4) (soup,12)
{soup,12}
(soup,12)
=====

=====SAVE=====
Success
=====

=====BTLOAD=====
Success
=====

=====BTLOAD=====
===== ERROR 200 =====
=====

=====PRINT_MIN=====
StandardItem      FrequentPattern      Frequency
soup -> {almonds} 2
soup -> {avocado} 2
soup -> {body spray} 2
soup -> {chicken} 2
soup -> {frozen vegetables} 2
soup -> {ground beef} 2
soup -> {hot dogs} 2
soup -> {milk} 2
soup -> {mineral water} 2
soup -> {protein bar} 2
soup -> {almonds, burgers} 2
soup -> {almonds, eggs} 2
soup -> {body spray, green tea} 2
soup -> {burgers, eggs} 2
soup -> {burgers, french fries} 2
soup -> {burgers, green tea} 2
soup -> {chocolate, ground beef} 2
soup -> {eggs, french fries} 2
soup -> {frozen vegetables, spaghetti} 2
soup -> {green tea, spaghetti} 2
soup -> {milk, spaghetti} 2
soup -> {almonds, burgers, eggs} 2
soup -> {burgers, french fries, green tea} 2
soup -> {burgers} 3
soup -> {chocolate, eggs} 3
soup -> {french fries, green tea} 3
soup -> {chocolate} 4
soup -> {eggs} 4
soup -> {french fries} 4
soup -> {spaghetti} 4
soup -> {green tea} 6
=====

```

```

=====PRINT_CONFIDENCE=====
StandardItem      FrequentPattern  Frequency      Confidence
soup -> {chocolate} 4 0.33
soup -> {eggs} 4 0.33
soup -> {french fries} 4 0.33
soup -> {spaghetti} 4 0.33
soup -> {green tea} 6 0.50
=====

=====PRINT_RANGE=====
StandardItem      FrequentPattern  Frequency
soup -> {burgers} 3
soup -> {chocolate, eggs} 3
soup -> {french fries, green tea} 3
soup -> {chocolate} 4
soup -> {eggs} 4
soup -> {french fries} 4
soup -> {spaghetti} 4
=====

```

□ 구현 시 반드시 정의해야하는 Class 및 멤버 변수

1. FPGrowth : FP-Growth 클래스

항목	내용	비고
threshold	빈도수 제한	멤버 변수로 변경 가능, 2 이상의 값을 가짐
fpTree	FP-Tree	자식 노드를 제외하고 NULL 값을 가짐
table	Header Table	
frequentPattern	생성된 Frequent Patterns	공집합과 집합의 크기가 1인 Frequent Pattern은 제외
fout	log 파일	
flog	result 파일	소수점 2자리까지만 입력 가능

2. FPNode: 아이템 정보 및 FP-Growth 노드

항목	내용	비고
item	상품 이름	
frequency	상품 빈도수	
parent	부모 노드	
next	다음 FPNode를 가리킴	
children	자식 노드를 가지고 있는 map 컨테이너	map<string, FPNode*> (상품명, 하위 상품 노드*)

3. HeaderTable: FP-Growth의 Header Table 클래스

항목	내용	비고
indexTable	상품 빈도수 및 상품명을 pair로 저장하고 있는 list	list<pair<int, string> > (상품의 빈도수, 상품 이름)
dataTabe	상품 정보를 가지고 있는 map 컨테이너	map<string, FPNode*> (상품명, 하위 상품 노드*)

4. FrequentPatternNode : Frequent Pattern 노드 클래스

항목	내용	비고
frequency	Frequent Pattern의 크기	
FrequentPatternList	Frequent Pattern들을 가지고 있는 multimap 컨테이너	multimap<int, set<string> > (Frequent Pattern 크기, Frequent Pattern)

5. BpTreeNode : B⁺-tree의 노드 클래스

항목	내용	비고
pParent	상위 BpTreeNode를 가리킴	
pMostLeftChild	하위 BpTreeNode 중 가장 왼쪽 노드를 가리킴	

6. BpTreeIndexNode : B⁺-tree의 인덱스 노드 클래스

항목	내용	비고
mapIndex	인덱스 값을 가지고 있는 map 컨테이너	map<int, BpTreeNode*> (빈도수, 하위 B ⁺ -Tree 노드*)

7. BpTreeDataNode : B⁺-tree의 데이터 노드 클래스

항목	내용	비고
mapData	해당 빈도수에 속한 Frequent Pattern들을 가지고 있는 map 컨테이너	map<int, FrequentPatternNode*> (빈도수, Frequent Pattern 정보 객체*)
pNext	다음 BpTreeNode를 가리킴	
pPrev	이전 BpTreeNode를 가리킴	

8. BpTree : B⁺-tree 클래스

항목	내용	비고
root	B ⁺ -tree의 루트	
order	B ⁺ -tree의 차수(m)	멤버 변수로 변경 가능

9. Manager : 다른 클래스들의 동작을 관리하여 프로그램을 전체적으로 조정하는 역할을 수행

항목	내용	비고
fpgrowth	FP-Growth	첫 번째 인자(threshold)를 받음
bptree	B ⁺ -tree	두 번째 인자(order)를 받음

□ Files

- ✓ market.txt : 프로그램에 추가할 장바구니 데이터가 저장된 파일
- ✓ result.txt : 연관 상품 집합들을 저장하고 있는 파일
- ✓ command.txt : 프로그램을 동작시키는 명령어들을 저장하고 있는 파일
- ✓ log.txt : 프로그램 출력 결과를 모두 저장하고 있는 파일

□ 제한사항 및 구현 시 유의사항

- ✓ 반드시 제공되는 코드(github 주소 참고)를 이용하여 구현하며 작성된 소스 파일의 이름과 클래스와 함

수 이름 및 형태를 절대 임의로 변경하지 않는다.

- ✓ 클래스의 함수 및 변수는 자유롭게 추가 구현이 가능하다.
- ✓ 제시된 Class를 각 기능에 알맞게 모두 사용한다.
- ✓ 프로그램 구조에 대한 디자인이 최대한 간결하도록 고려하여 설계한다.
- ✓ 채점 시 코드를 수정해야 하는 일이 없도록 한다.
- ✓ 주석은 반드시 영어로 작성한다. (한글로 작성하거나 없으면 감점)
- ✓ 프로그램은 반드시 리눅스(Ubuntu 18.04)에서 동작해야 한다. (컴파일 에러 발생 시 감점)
 - 제공되는 Makefile을 사용하여 테스트하도록 한다.
- ✓ 채점은 제공된 파일이 아닌 다른 1000개 이상의 구매 상품 목록을 사용하여 진행된다.
- ✓ FP-Growth는 FP-Tree와 HeaderTable 연결까지가 채점 기준이고 result.txt는 FP-Growth의 결과물로 result.txt 생성까지 구현할 경우 가산점이 제공된다. 구현을 못 한 학생들을 위해 따로 result.txt를 제공한다.
- ✓ 제공되는 testcase 파일과 result 파일은 각각 market.txt와 result.txt로 변경해서 사용한다.
 - ※ (testcase1, result1), (testcase2, result2), (testcase3, result3)

□ 제출기한 및 제출방법

- ✓ 제출기한
 - 2020년 11월 6일 금요일 23:59:59 까지 제출
- ✓ 제출방법
 - 소스코드(Makefile과 텍스트 파일 제외)와 보고서 파일(pdf)을 함께 압축하여 제출
 - 확장자가 .cpp, .h, .pdf가 아닌 파일은 제출하지 않음
 - 보고서 파일 확장자가 pdf가 아닐 시 감점
 - 제출 사이트는 추후 공지
- ✓ 제출 형식
 - 학번_DS_project2.tar.gz (ex. 2020XXXXXX_DS_project2.tar.gz)
- ✓ 보고서 작성 형식 및 제출 방법
 - Introduction : 프로젝트 내용에 대한 설명
 - Flowchart : 설계한 프로젝트의 플로우차트를 그리고 설명
 - Algorithm : 프로젝트에서 사용한 알고리즘의 동작을 설명
 - Result Screen : 모든 명령어에 대해 결과화면을 캡처하고 동작을 설명
 - Consideration : 고찰 작성
 - 위의 각 항목을 모두 포함하여 작성
 - 보고서 내용은 한글로 작성
 - 보고서에는 소스코드를 포함하지 않음