**Date:** 2/13/2018

**Technical Assignment 2**

**IS 665 Spring 2018**

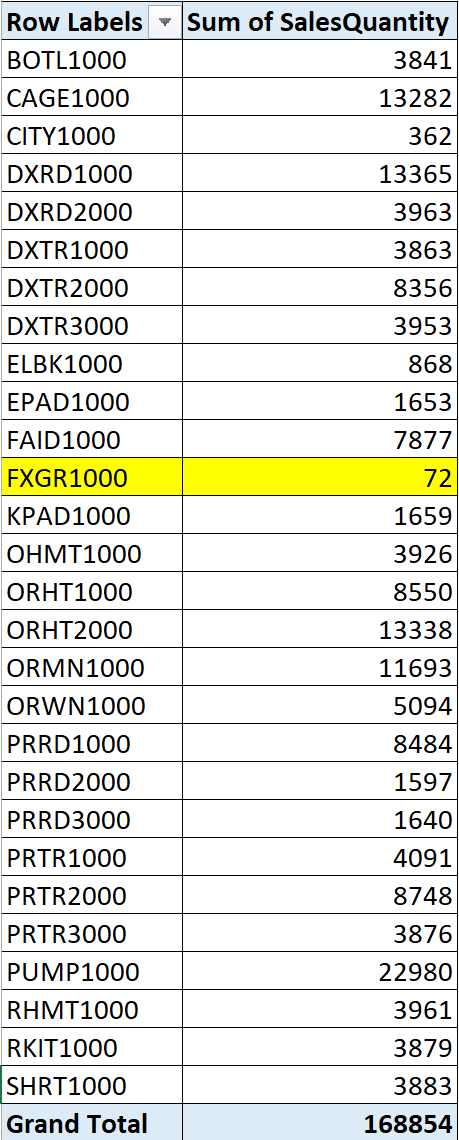
Richa Bharadwaj

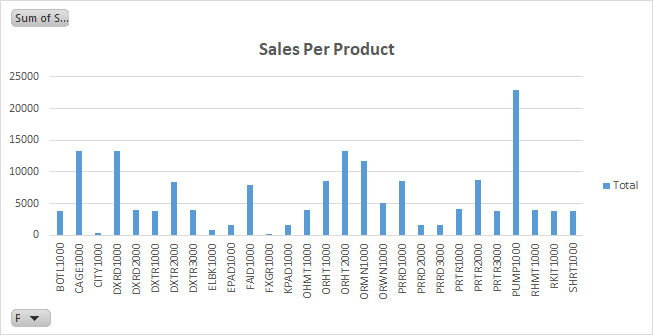
**Part I. Statistics (60 pts.)**

At the end of lab I, I asked the following questions:

### **1.** **Flops**

What product sold the least number of units?



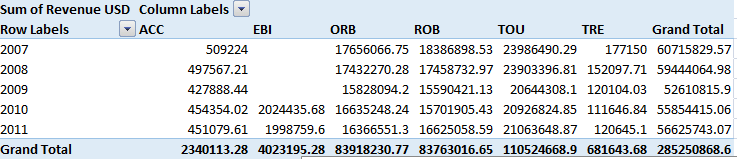


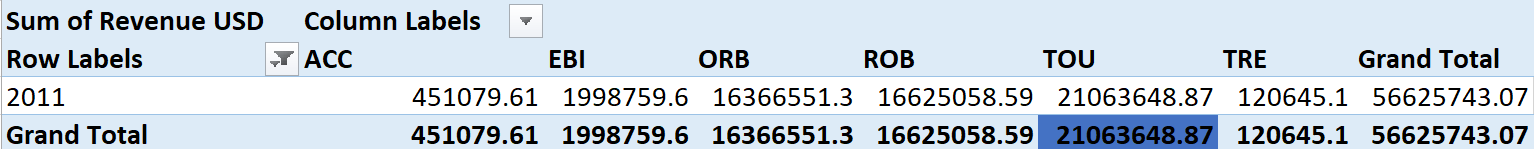
**Conclusion :** The above pivot table and graph shows that **FXGR1000 (Fixed Gear Bike Plus)** sold the least number of units.

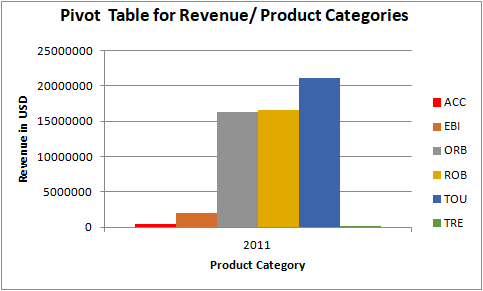
--------------------------------------------------------------------------------------------------------------------------------------

### **2.** **Top Seller**

What product category provided the most revenue in 2011?







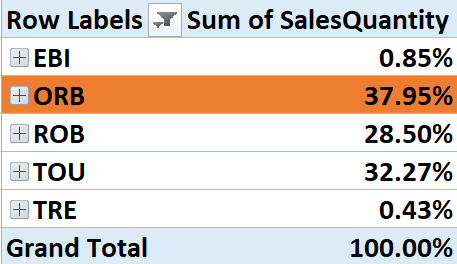
**Conclusion:**

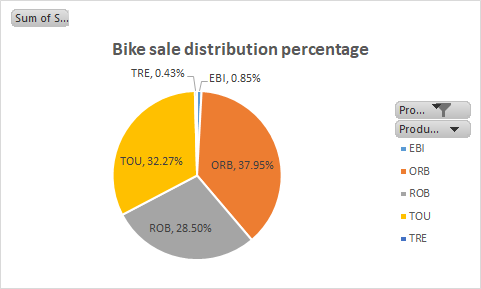
* From Pivot table it is evident that Product category - **TOU** marks the highest revenue of **21063648.87 USD** in the year 2011. This is almost equal to 37% of total gross.
* TOU covers 6 product types (DXTR1000,DXTR2000,DXTR3000,PRTR1000,PRTR2000,PRTR3000) amongst which DXTR2000 - Deluxe Touring Bike (silver) contributes the highest counting for 143602.6 USD (0.68% of gross total).
* Overall Product category - TOU has maintained consistency in its revenue through the years 2007 to 2011 and always stayed above average range

--------------------------------------------------------------------------------------------------------------------------------------

### **3.** **Sales by Product Category**

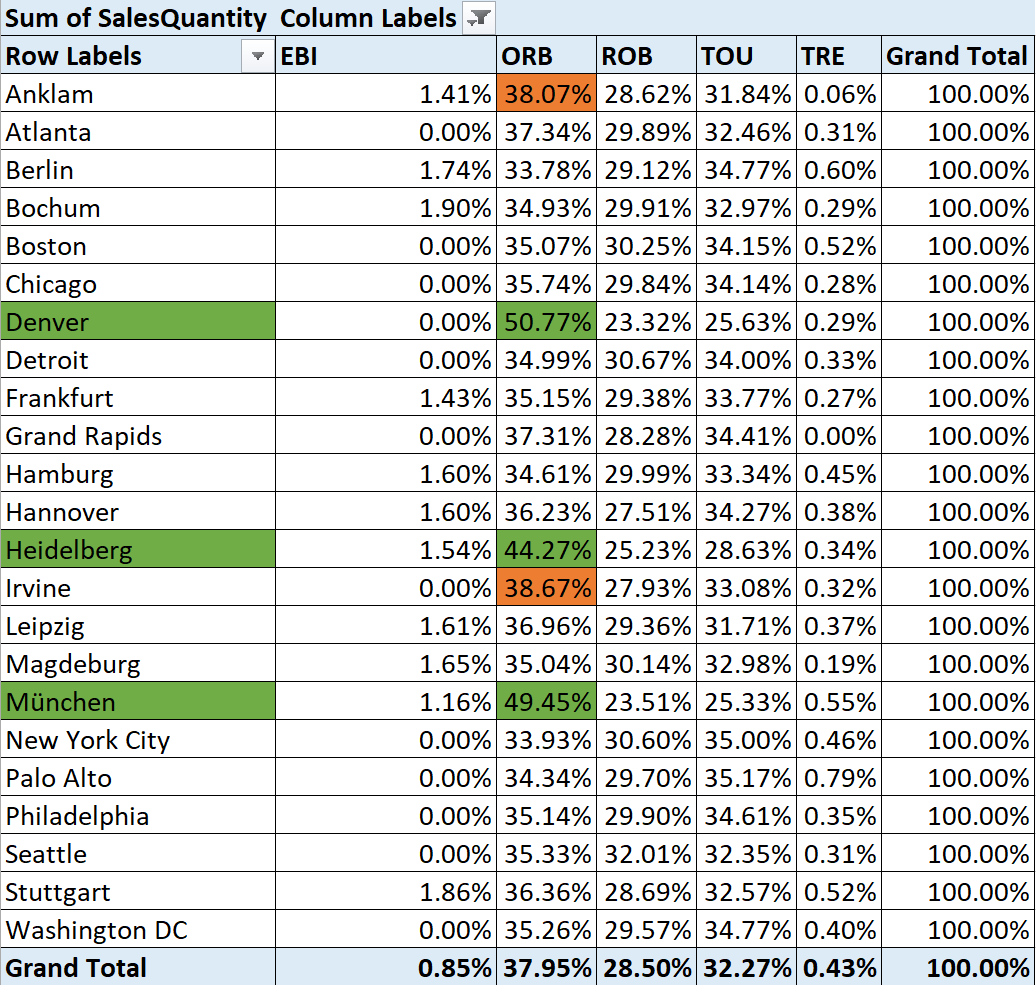
**a)What percentage did the off-road bikes contribute to the overall bicycle sales quantity?**

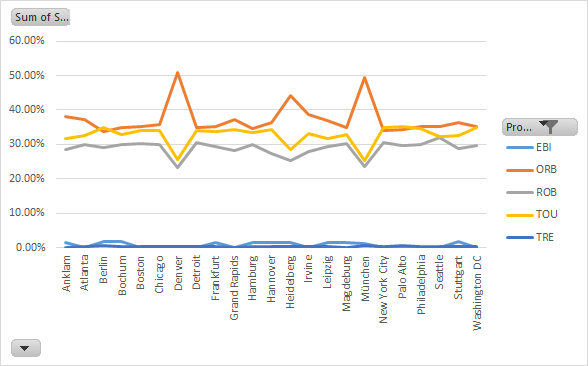


****

**Conclusion :** The above pivot table and the pie chart shows that the overall contribution of Off-Road Bikes(**ORB**) to the total Sales Revenue is **37.95%**. (Here ACC is filtered out as it doesn’t belong to the Bike category).

**b) In which three cities was this percentage significantly above the average?**

****

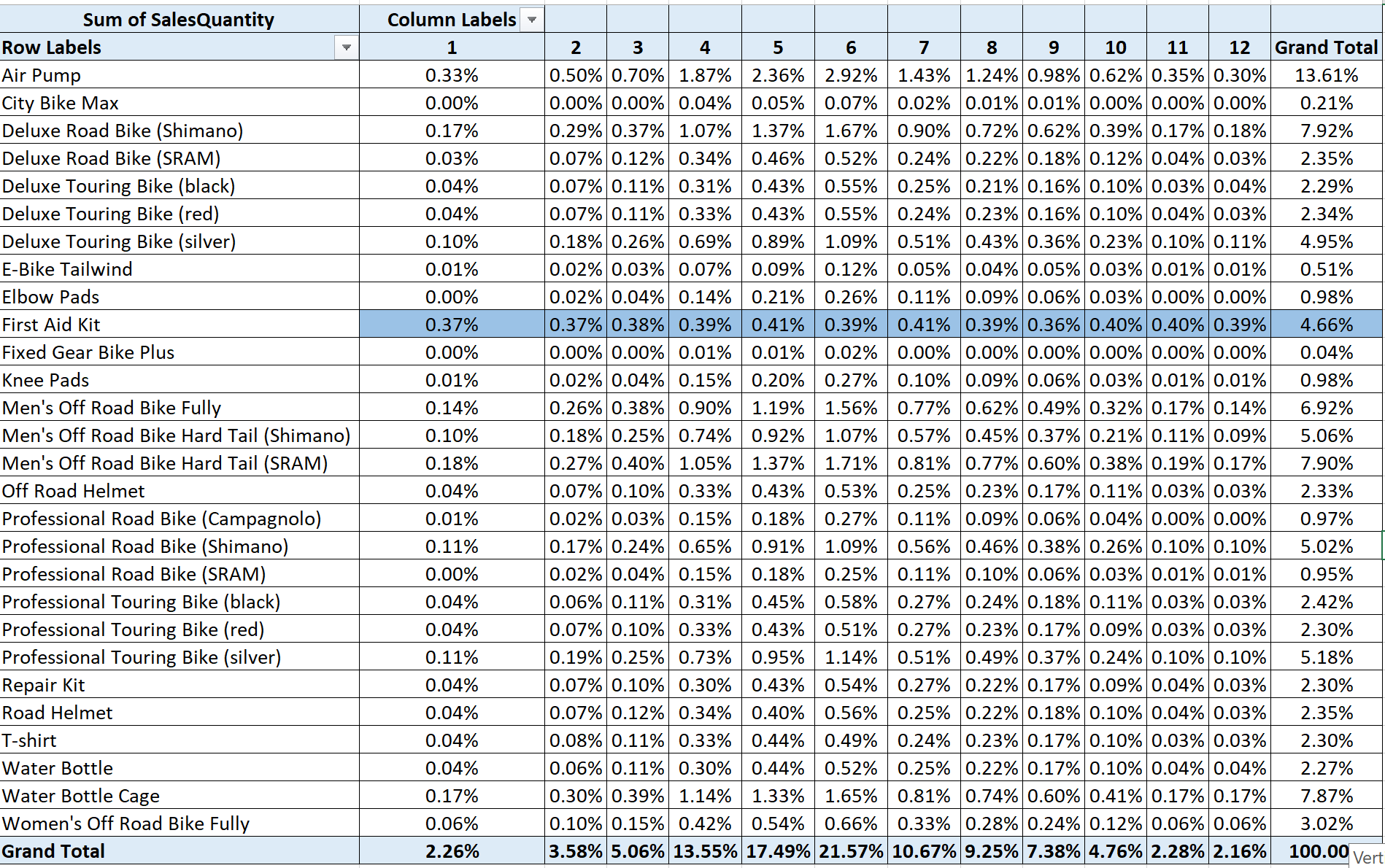
****

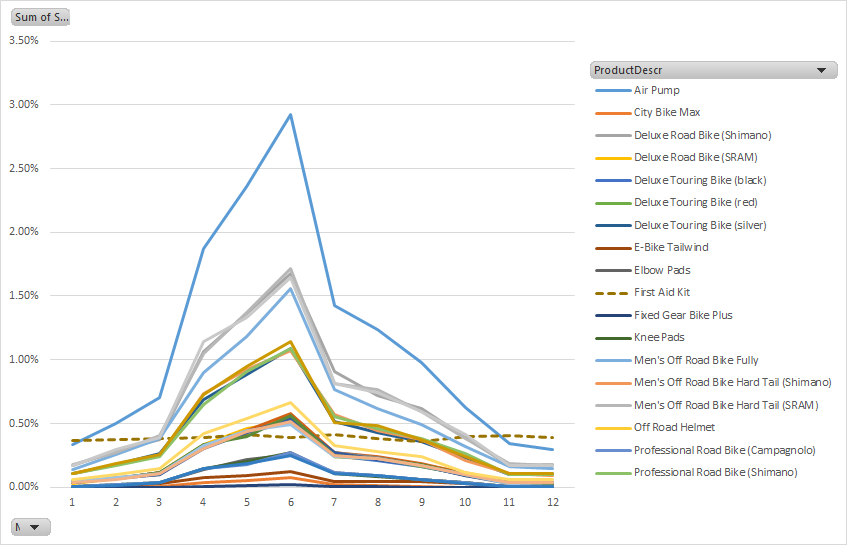
**Conclusion :** In the above pivot table (filtered Product category and converted to data to Row %), the ORB grand total is **37.95%**.The above avg (37.95%) values are highlighted with RED color., which narrowed down to five cities and then Top 3 cities are highlighted with **GREEN** color. That gave us cities like **Denver, Heidelberg and München** as the cities which contributed above avg. to the percentage of ORB’s overall contribution. The same is depicted by the line graphs (which shows clear spikes for these 3 cities).

**--------------------------------------------------------------------------------------------------------------------------------------**

**4.** **Seasonal Behavior**

Bicycles and accessories are more likely to be bought in spring and summer as in fall and winter time. Find which product is an exception to this rule.





**Conclusion :** As per the above pivot table and graph ,all the products sales are high in summer and its fluctuating on the other months. Whereas **FAID1000 (First Aid Kit)** is a clear exception to the rule.Its sales remains more or less constant throughout the year. Line graphs ( **FAID1000 is shown as dotted lines**) and also towards winter it is clearly selling higher than other products.

Please develop your pivot table to answer the above questions. FOR EACH QUESTION, SHOW A SCREENSHOT OF YOUR PIVOT TABLE, THEN SHOW YOUR ANALYSIS BASED ON THE SCREENSHOT AND YOUR CONCLUSION.

Your grade is not only based on your screen shot, but also on how thorough your analysis is, and whether your analysis is consistent with the pivot table your constructed.

**Part II. Data Structure (40 pts.)**

In today’s class, we reviewed the data structure called stack and queue, and learned about BFS and DFS.

In the following graph, nodes are represented as alphabets.

A ---- B ---R ---K

/ / /

/ / /

E ---- F ---P

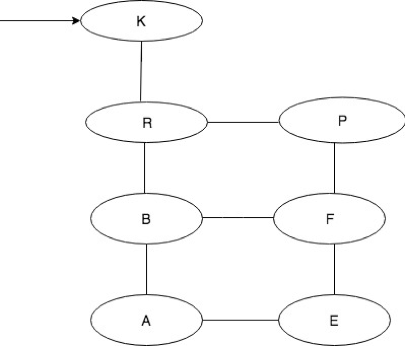
**Question 1.** Please perform a Depth-First-Search (DFS) to find all paths from K to E. Please draw a tree to show your process. (20 points)

**Solution 1.**

* Initially, starting from point K being the parent class. So we will update it in the stack Table and same for the output.
* So the stack table has K and output too has K

STACK Table Output --- K

|  |
| --- |
| **K** |



Next we are going to consider R being the current node as it is adjacent to the K node. We will point towards the child class R.

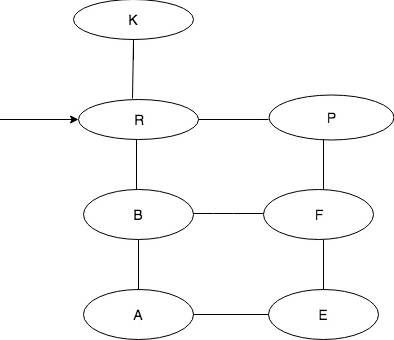
STACK Table Output--- K R

|  |
| --- |
| R |
| K |

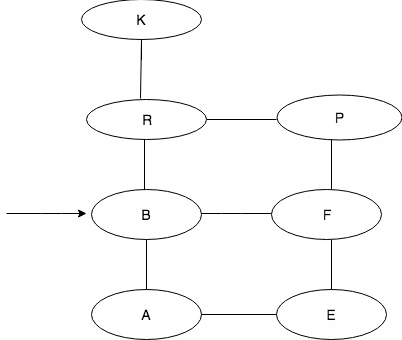
* Since it is DFS and we have to consider it alphabetically we will take B as the current node and mark it as visited.

STACK table Output---- K R B

|  |
| --- |
| B |
| R |
| K |



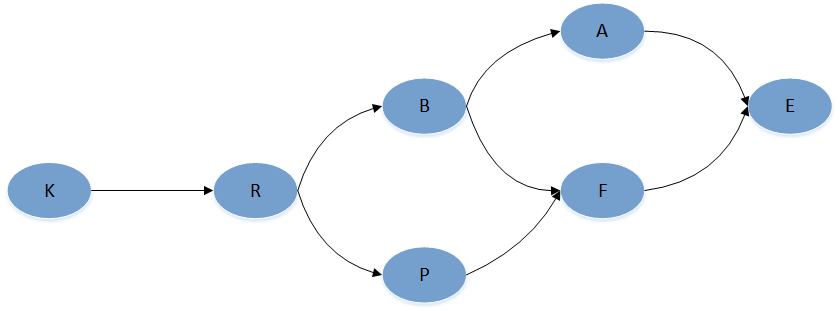
* From R next to K continues to search. Select B to continue search, then select A next to B to continue, finally get E.



* To reach E we have to either consider F or A. As A being closest in the alphabetically order we are going to take A as the current node.
* Come back to A to search is there any other node which connect to A. come back to B to search is there any other node which has not been accessed.
* We found F, so from F to search, we get E.
* Come back to former level F, we look for is there any other node which has not been accessed. We continue to come back to B, also look for any other node which has not been accessed.
* Come back to R, we look for any other node which has not been accessed, we get P.;
* From P we search for node next to P, we get F. we continue to search node, finally get E. The stack table has these details. Using this Left- First approach this is the output:

Stack Table Final Output: K-R-B-A-E

|  |
| --- |
| E |
| A |
| B |
| R |
| K |



In the end, we get three paths to find E from K.

**K-R-B-A-E---- Left-First Approach**

**K-R-P-F-E----Right-First Approach**

**K-R-B-F-E----- other possibility**

**Question 2.**  Please perform a Breadth-First-Search (BFS) to find all paths from K to E. Please draw a tree to show your process. (20 points)

* **Solution 1.** Considering the queue table for BFS. We will begin from the node K being the parent node.
* Then we will move to its next child node which is R. Adjacent to R is B and P. Now we would mark the nodes B R and P as visited.

Queue Table Output -- K R B P

|  |
| --- |
| P |
| B |
| R |
| K |

* Now we will consider adjacent nodes to P and B which is F and A. We would mark B as dequeue.
* Since A comes first in alphabetical order we would consider A in the queue table and then F and mark them as visited.

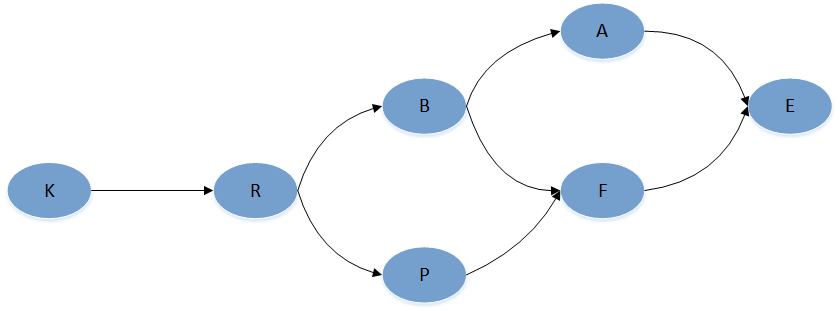
Queue Table Output-- K R B P A F

|  |
| --- |
| F |
| A |
| P |
| B |
| R |
| K |

* We will travel back to A and F and see any further way to come to node E. Since E is the child class and adjacent to A and F. We will travel from F to E. This is the final output using the Left-First approach

Queue Table **Final Output: K-R-B-P-A-F-E**

|  |
| --- |
| E |
| F |
| A |
| P |
| B |
| R |
| K |



**K-R-B-P-A-F-E -----using the Left-First Approach**

There can be other possibilities as well.

**K-R-B-P-F-A-E ----- using the Right-First Approach**

**K-R-P-B-F-A-E**

**K-R-P-B-A-F-E**

**HINT: If logic alone does not help, you could pretend to be a “human stack” or “human queue”, follow the algorithm and pseudo code we covered in class and get the results.**

**EXCEL ATTACHED-PART 1**

