University of Nevada, Las Vegas Computer Science 477/677 Fall 2021

Answers to Preparation for the Second Test Monday October 18, 2021

1. I will ask questions on the materical in the fourth and fifth assignments. and also from the following material of this document.

2. One of the most commonly used search structers is unordered list. Even though you need linear search to find an item, the simplicity of the structure makes it useful for small sets of data.

   A *self-organizing list* is a list where items are moved closer to the head if they are more frequently used. One standard method of self-organization is called *move to front*, where an item which is found is moved to the front of the list. This method is predicated on the "locality" assumption, that is, that items are not sought equally often, and it thus efficient to assume that if an item is sought, it is likely to be sought again.

   Suppose that items are letters, and the list is QWERTYUIOP. When we execute find(I) the list becomes IQWERTYUOP.

   There is a more advanced method, called BIT. Initially, a random bit is placed at each item. When find(X) is executed, the bit is flipped, and item X is moved to the front if and only if its bit is 1,

   For example, assume the list (with its bits) is initialized to be

   ```
   0 1 1 1 0 1 0 0 0 1
   Q W E R T Y U I O P
   ```

   If we request find(T) followed by find(E), the list becomes

   ```
   1 0 1 0 1 1 0 0 0 1
   T Q W E R Y U I O P
   ```

   Here is another example. If the initial list is

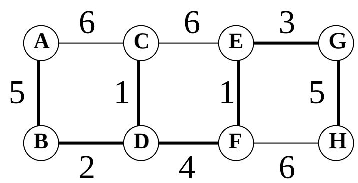
   ```
   0 1 1 1 0 1 0 0 0 1
   Q W E R T Y U I O P
   ```

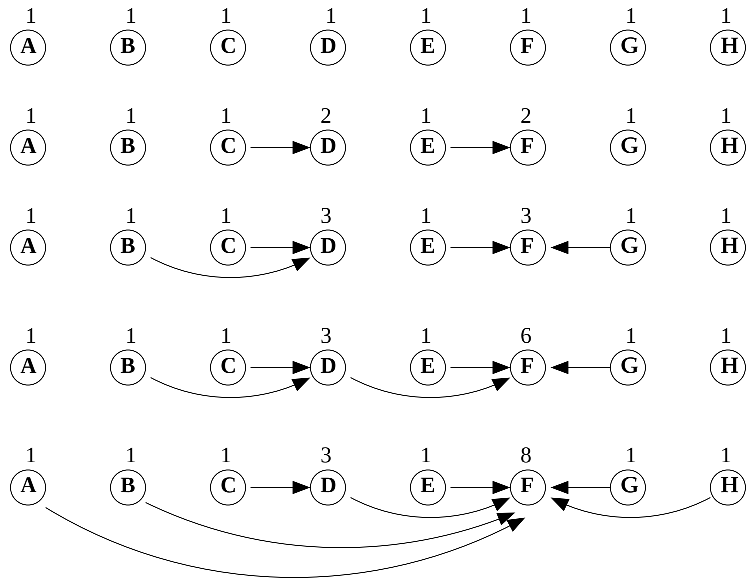
   what is the list, with its bits, after executing these operations in this order?

   find(R);
   find(E);
   find(P);
   find(E);
   find(I);

   ```
   0 1 1 0 0 1 0 0 0 1
   Q W E R T Y U I O P

   0 1 0 0 0 1 0 0 0 1
   Q W E R T Y U I O P

   0 1 0 0 0 1 0 0 0 0
   Q W E R T Y U I O P

   1 0 1 0 0 1 0 0 0 0
   E Q W R T Y U I O P

   1 1 0 1 0 0 1 0 0 0
   I E Q W R T Y U O P
   ```

3. Walk through Kruskal's algorithm to find the minimum spanning tree of the weighted graph shown below. Show the evolution of the union/find structure at several intermediate steps. Whenever there is choice between two edges of equal weight, choose the edge which has the alphabetically largest vertex. Whenever there is a union of two trees of equal weight, choose the alphabetically larger root to be the root of the combined tree. Indicate path compression when it occurs.





In the last step, find(A) and find(B) are executed. Execution of find(B) causes path compression, changing the parent of B from D to its leader F.

4. Construct an AVL tree by starting with en empty tree and then inserting the letters B,A,E,D,F,C in that order.

You can insert B,A,E,D,F without any rebalancing. But when you insert C, you will need to rebalance. Do you need a double rotation?

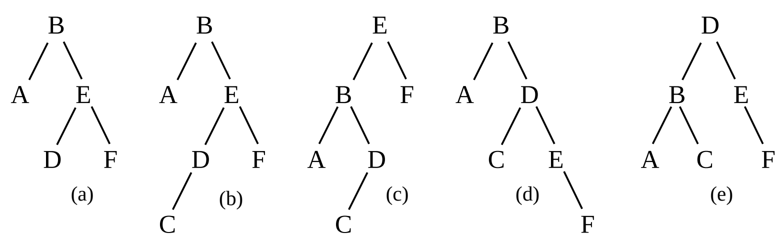


Figure (a) shows the AVL tree before insertion of C. Figure (b) shows the tree after insertion of C, and it is clearly unbalanced. Figure (c) shows the effect of a simple left rotation at B. The tree does not become balanced, therefore we need a double left rotation at B, consisting of a right rotation at D, yielding the tree in Figure (d), followed by a left rotation at B yielding the final balanced tree shown in Figure (e).

5. Let A be a triangular array $\{A[i,j]\}$ where $0 \le i \le j \le 5$. How many items are in the array? (Answer: 21) If A is stored in RAM in row-major order, with base adress 1024, where is $A[2,3]$ stored? Assume each item uses one address of RAM.

   We are not interested in the values of of A, only in the offset of each position. The easiest way to see those offset values is to sketch a triangular array:

	0	1	2	3	4	5
0	0	1	2	3	4	5
1		6	7	8	9	10
2			11	12	13	14
3				15	16	17
4					18	19
5						20

   The entry in row 2 column 3 is 12, hence that is the offset. Thus $A[2,3]$ is stored in position $1024+12 = 1036$ of the RAM.