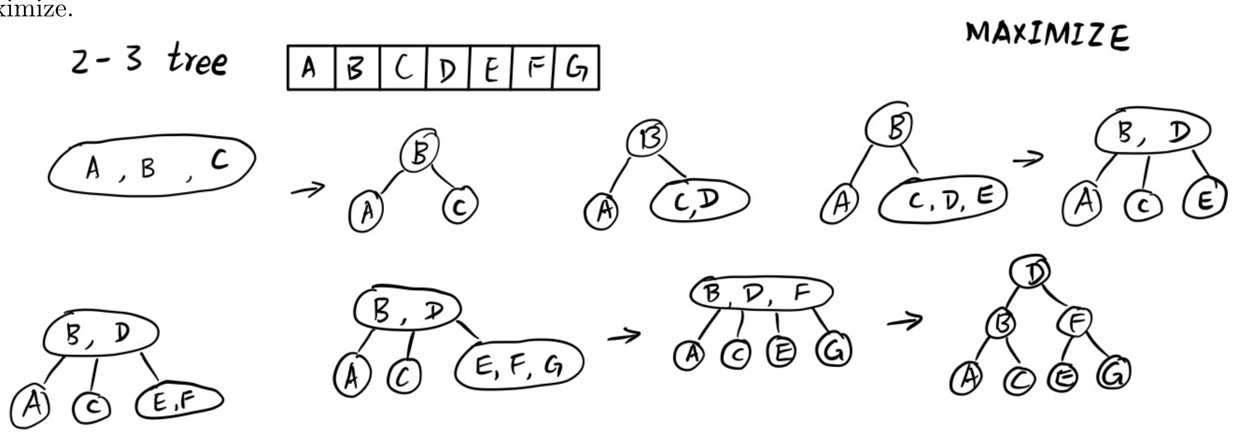
PROBLEM SET 2, WRITTEN PART
Balanced Binary Search Trees and MSTs

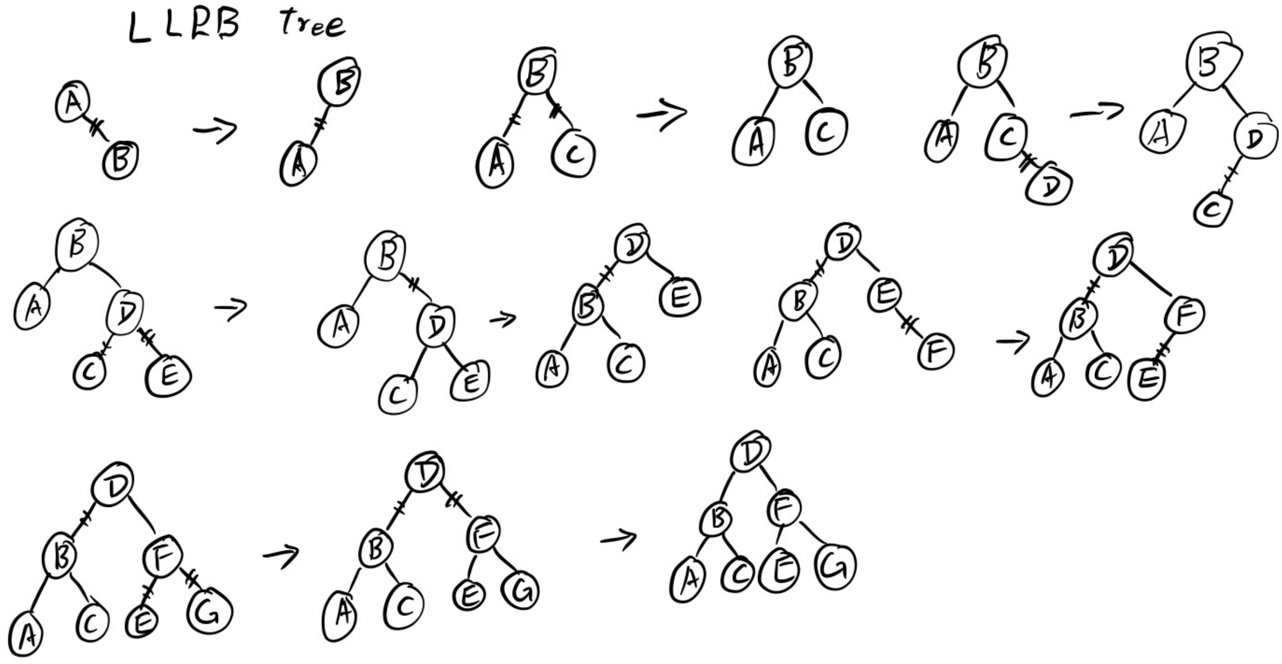
Wenbo, Wu

October 24, 2019

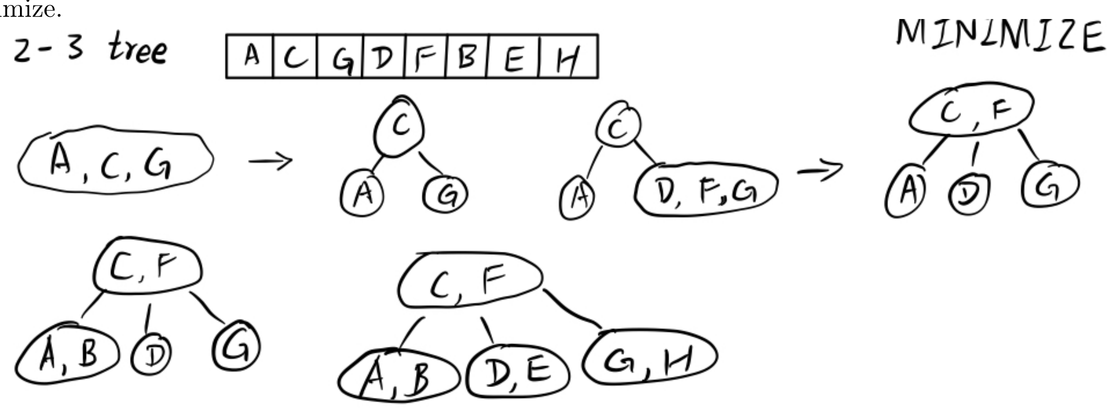
1 Number of nodes and balance

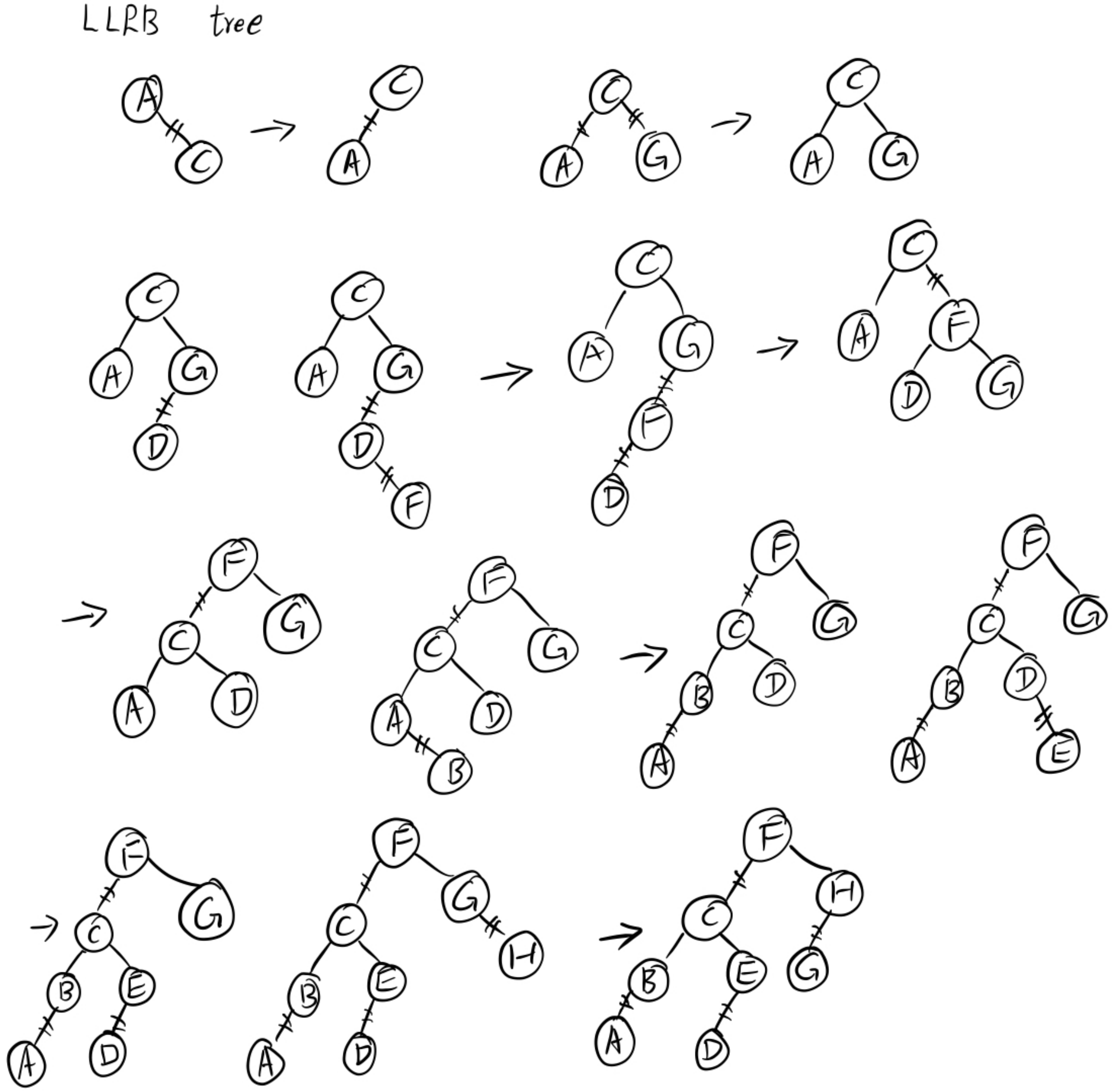
Maximize.





Minimize.





The final result for 2-3 tree is the similar as the LLRB tree. They both are black balance for LLRB tree and 2-3 tree. In other words, the black line height in LLRB tree is the same as the height of 2-3 tree. The different level of balance for LLRB tree is 1. The longest root-to-leaf path and the length of the shortest root-to-leaf path the differ of height maximum is 1.

2 A hybrid MST algorithm

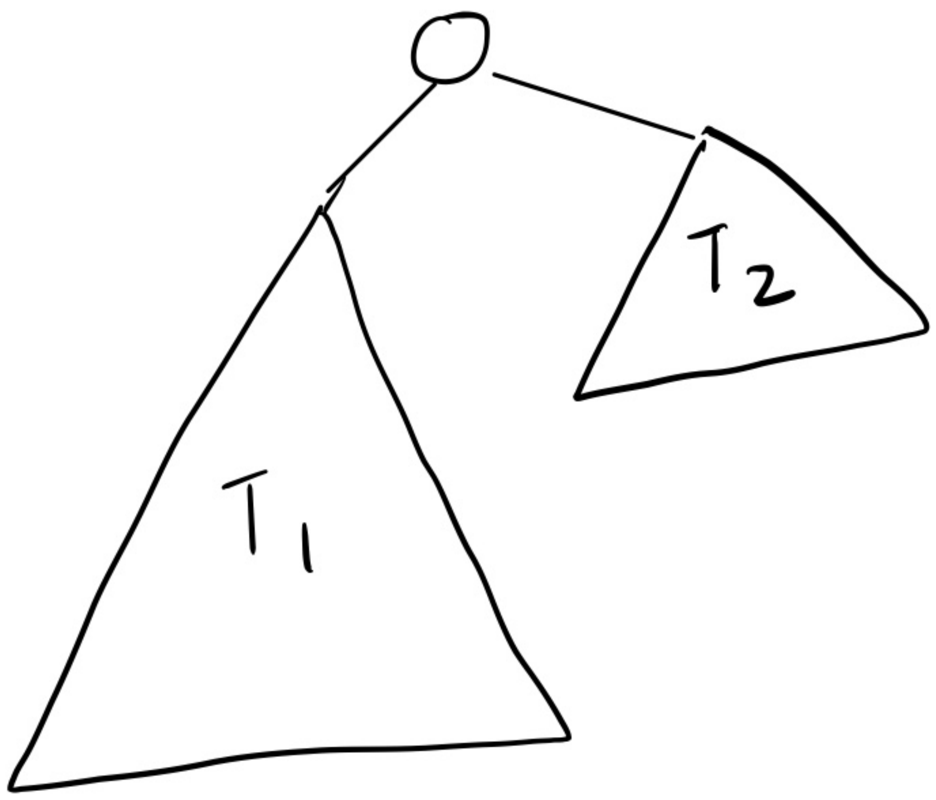
Yes, it work.
Since, when $\{A\}$ = even. Running Kruskal's step which adding e. e is the least cost e. When $\{A\}$ = odd. Prim's Step adding the smallest edges that connect with the spanning tree. Let T be the spanning tree found by Hybrid MST algorithm and T* be the MST of the graph. Assume T $\neq$ T*. Therefore, T-T* $\neq \emptyset$. Let (u,v) be any edge in T-T*. When (u,v) was add to T, it was a least-cost edge crossing some cut(S, V/S). Since T* is an MST there must be a path from u to v in T*. This path begins in S and ends in V/S, so there must be some edge(x,y) along that path where x $\in$ S and y $\in$ V/S. Since (u,v) is a least-cost edge crossing(S,V/S), we have c(u,v) < c(x,y). We T*' be the replace (x,y) with (u,v) spanning tree. The weight of C(T*') < c(T*). Which contradicting the T* is an MST. So T is an MST.

3 Uniqueness of MSTs

Proof by contradiction: Assume that there are two distinct minimum spanning trees, A and B. Consider the edge e of minimum weight among all the edges that are contained in exactly one of A and B. Without loss of generality, this edge appears only in A we call it e1. Then B $\cup$ e1 must contain a cycle, and one of the edges of this cycle, call it e2, is not in A. Since e2 is a edge different from e1 and is contained in exactly one of A or B, it must be that w(e1) < w(e2). Note that T = B $\cup$ e1/e2 is spanning tree. The total weight of T is smaller than the total weight of B, but it is contradiction, since we have supposed that B is a minimum spanning tree.

*** Optional extra credit question: Non-distinct edge weights ***

4 Quick-Union with Union-by-Rank



$$h_1 \leq \log_2 |T_1| \leq \log_2 n$$

Initially when each node is the root of its own tree, it's trivially true. The case when the rank of a node might changed is when the Union by Rank operation is applied. The tree with smaller height T_2 will be attached to T_1 with a greater height. And all nodes visited along the path will be attached to the root, which has larger rank than its children. In the worst case, the h_1 upper bound would be $\log_2 S_1$ where S_1 is the number of elements in T_1 tree. For the worst case is that the T_2 tree only contain one elements and T_1 contains n-1 elements. The upper bound $log_2 T_1 \leq log_2 n$. Therefore, the worst case run time is O($log_2 n$).