Hw14:

Thought of experiment and conclusion:

1. Union same set of numbers in same order by two rules(height rule and weight ruls)
2. Collapsing Find the leaf nodes.
3. Compare the height/weight of roots and the union array before and after collapsing find.
4. Conclusion:

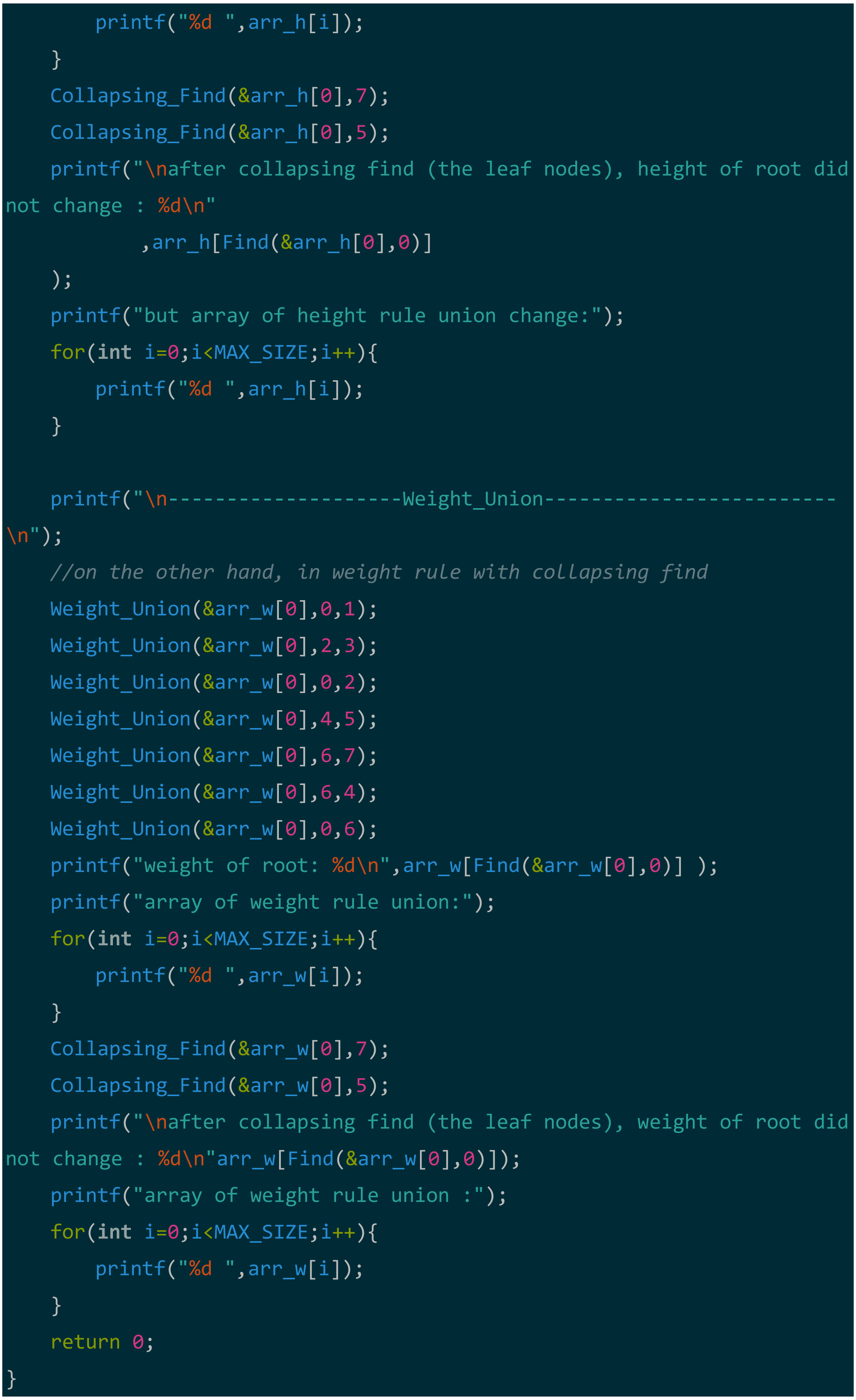
   Collapsing find is better for weight rule.

   Because, after collapsing find, the root weight did not contradict to the content of the union array(collapsing find won’t affect the weight of tree). However, in height rule, the height may change after the collapsing find, but it is hard to update the height information of root. Thus, weight rule union is better in conjunction with collapsing find.

Experiment:

```
int main(){
    int arr_h[MAX_SIZE];
    int arr_w[MAX_SIZE];
    for(int i=0;i<MAX_SIZE;i++){
        arr_h[i]=-1;
        arr_w[i]=-1;
    }
   printf("\n--------------------Height_Union-------------------------\n");

    //set with height =4
    Height_Union(&arr_h[0],0,1);
    Height_Union(&arr_h[0],2,3);
    Height_Union(&arr_h[0],0,2);
    Height_Union(&arr_h[0],4,5);
    Height_Union(&arr_h[0],6,7);
    Height_Union(&arr_h[0],6,4);
    Height_Union(&arr_h[0],0,6);
    printf("height of root: %d\n",arr_h[Find(&arr_h[0],0)] );
    printf("array of height rule union:");
    for(int i=0;i<MAX_SIZE;i++){
```



```
        printf("%d ",arr_h[i]);
    }
    Collapsing_Find(&arr_h[0],7);
    Collapsing_Find(&arr_h[0],5);
    printf("\nafter collapsing find (the leaf nodes), height of root did not change : %d\n"
            ,arr_h[Find(&arr_h[0],0)]
    );
    printf("but array of height rule union change:");
    for(int i=0;i<MAX_SIZE;i++){
        printf("%d ",arr_h[i]);
    }

    printf("\n--------------------Weight_Union-------------------------\n");
    //on the other hand, in weight rule with collapsing find
    Weight_Union(&arr_w[0],0,1);
    Weight_Union(&arr_w[0],2,3);
    Weight_Union(&arr_w[0],0,2);
    Weight_Union(&arr_w[0],4,5);
    Weight_Union(&arr_w[0],6,7);
    Weight_Union(&arr_w[0],6,4);
    Weight_Union(&arr_w[0],0,6);
    printf("weight of root: %d\n",arr_w[Find(&arr_w[0],0)] );
    printf("array of weight rule union:");
    for(int i=0;i<MAX_SIZE;i++){
        printf("%d ",arr_w[i]);
    }
    Collapsing_Find(&arr_w[0],7);
    Collapsing_Find(&arr_w[0],5);
    printf("\nafter collapsing find (the leaf nodes), weight of root did not change : %d\n"arr_w[Find(&arr_w[0],0)]);
    printf("array of weight rule union :");
    for(int i=0;i<MAX_SIZE;i++){
        printf("%d ",arr_w[i]);
    }
    return 0;
}
```

```
---------------------------------------------------------
Output:
--------------------Height_Union------------------------
height of root: -4
array of height rule union:-4 0 0 2 6 4 0 6 -1 -1 -1 -1 -
1 -1 -1 -1 -1 -1 -1 -1
after collapsing find (the leaf nodes), height of root
did not change : -4
but array of height rule union change:-4 0 0 2 0 0 0 0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
--------------------Weight_Union------------------------
weight of root: -8
array of weight rule union:-8 0 0 2 6 4 0 6 -1 -1 -1 -1 -
1 -1 -1 -1 -1 -1 -1 -1
after collapsing find (the leaf nodes), weight of root
did not change : -8
array of weight rule union :-8 0 0 2 0 0 0 0 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1
```

Explanation:

With height rule union, after the ***collapsing find*** of number 7 and 5, the height of root should change from -4 to -3. However, it is difficult to trace the height of tree top-down in union array (the connections are only in bottom-up direction).

On the other hand, the ***collapsing find*** won’t change the weight of any tree, so it is better to use weight rule union in conjunction with ***collapsing find***.