GEBZE TECHNICAL UNIVERSİTY

COMPUTER ENGINEERING

COMPUTER ORGANIZATION

CSE 331 – 2019 FALL

HW1 REPORT

SILA BENGİSU YÜKSEKLİ

1801042877

➢ *Explanation of Codes*

This solution is can be used with only six elements that are in union set, it can not be used with more or less elements and you can't enter zero or negative numbers for a reason that I will explain later. Because I allocate 24 byte in data section.(Every integer has four bytes and there are six elements.) You can work with any number of subsets. You can allocate any size you want for each subset. According to the moodle page, I limit number of subsets ten and size of each subset twenty. So I allocate 900 byte in data section, but if you want to work with more number and size, you can change the values located at data section. Actually, size of the union set can be incrased too, but in this situation, code must be changed. For example, I stored every element that belongs to union set, in $s1, … , $s6 register. That's why, some code lines like this, has to be changed.

In text section, first I read the elements of the union set from user:

```
    li   $t1, 0
    li   $t3, 0
takeElements:
    beq  $t1, 6, endOfTaking
    …
    move $t2, $v0
    sw   $t2, UnionSet($t3)
    …
```

$t1 is the loop counter, if $t1 is equal to six finish the loop. Because I work with 6 elements. $t3 register indicates array index of UnionSet array. I move elements that I read from keyboard to $t2 register. Then, I store each element in UnionSet array

After that, I read the number of subsets from user and store this value in $t0 register. Then, there are two loops, one of them is nested. Nested loop continues until loop counter reaches the size of the subset you entered. Outer loop continues until loop counter reaches the number of the subsets you entered:

```
    li   $t5, 0
    li   $t1, 0
biggerLoop:
    beq  $t1, $t0, biggerLoopEnd   .
    …
    li   $v0, 5
    syscall
    move $t3,$v0
    sw   $t3, SubSets($t5)
    addi  $t5, $t5, 4
   li   $t4, 0
  nestedLoop:
    …
```

There are some parts of the code block that explained above. $t5 register represents array index of the SubSets array. $t1 register is loop counter. If it reaches number of the subset, loop will be ended. In this Subsets array , I also added the size as the first element of the array and index of subset as the second element of SubSets array. For example, user want to enter 2 subsets, One of them has size which is one, second one has size which is two. Then array will be like this: {1,0,2},{2,1,3,4}

After loop ends, I make a reference of Union set and I store every element in in $s1, …. $s6 register. Then greedy algorithm begins. The outermost loop which is called whileLoop checks whether the size of UnionSet array reaches zero. If it reaches, it means all the elements in union set,are choosen. $t1 register is loop counter of the whileLoop. $t2 register represents the index of the IndexArray that stores the indexes of the subsets that is choosen.

$t3 register is the loop counter of the whileSubset loop. $t4 is the loop counter of whileSize loop. $t5 register represents maximum number. The initial value is zero. When loop is started, code will check every subset whether which one most covers the UnionSet array, then according to the value that is found, it changes max value($t5).

If $t3 loop counter reaches number of the subsets(This value is stored at $t0 register), this loop will be end. $t7 register which has initial value as zero, is counter for number of elements that covers most. Then this value will be compared with max number($t5).

In nested loop wich is called whileSize, compares value located at current index with values in register$s0, …, $s6. According to comparison, it may increase value at $t7 register. When whileSize loop ends, it checks the number at $t7 register. If value at $t5 register is smaller than value at $t7 register, it changes the max number. In the if block, it also stores the beginning index and last index of the elements, that belongs to subset which covers the UnionSet array most, (index is type of byte : 0,4,8,..) at $a1 and $a2 registers.

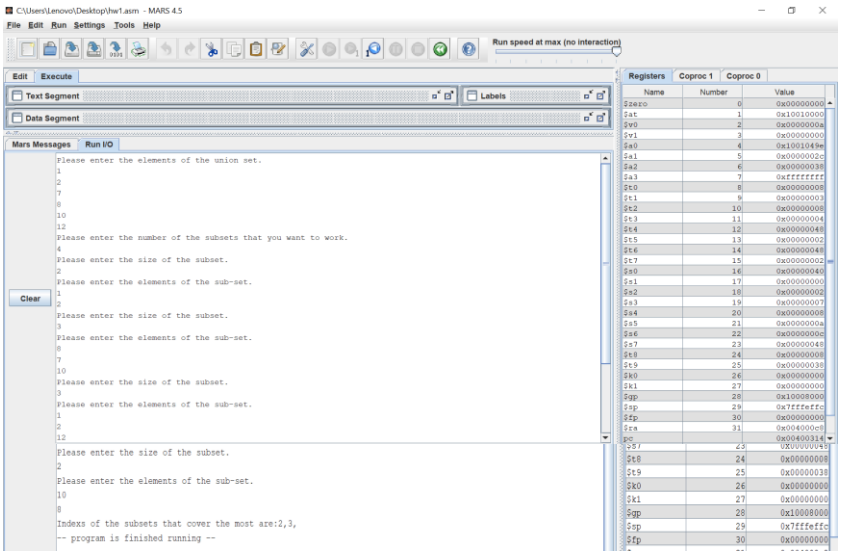
When whileSubset loop is ended, I save the index of the subset that is choosen, in IndexArray. Beginning index of the subset that is choosen, is begins at third element. Because first element is size value and second element is index value. So, to store index value at IndexArray, I copy the value at $a1 register to $t9 register and subtract four from $t9 register to reach the right index which has index of the subset.

Then, there is one loop that is called whileEliminateNumbers. I copy beginning index to $t9 register. If beginnig index reaches last index value located at $a2 register, loop will be end. The aim of this loop is that: it makes all elements that is chosen, -1. For example UnionSet has 6 elements which are 1,2,3,4,5,6 and let's say there is a subset which has elements like this: 1,2,3,4,5. So I remove this values and changed with -1 not to look again in next iteration.

There is another loop at the end which is called whileEliminateUnionSet. The aim of this loop is that: If elements that is choosen, are same as the elements in UnionSet, I remove this values from UnionSet and decrease the value at $t1 register and it returns back to the outermost loop which is whileLoop. This loop continues until all the elements in UnionSet, are removed.

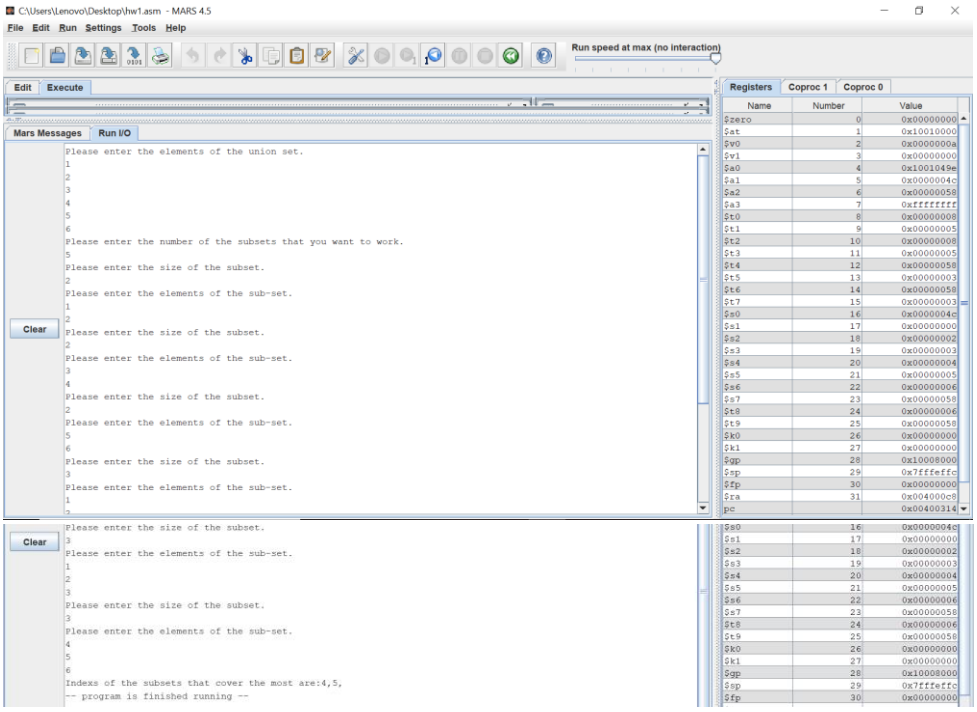
- ***Test Cases***
  1. ***Test Case 1***

     In this case, elements of the UnionSet are 1,2,7,8,10,12. There are four subsets. First has 1,2. Second has 8,7,10. Third has 1,2,12. Last one has 10,8. As you can see union of second and third subset gives us UnionSet. I can choose another combination of course, but this case has the least numbers.



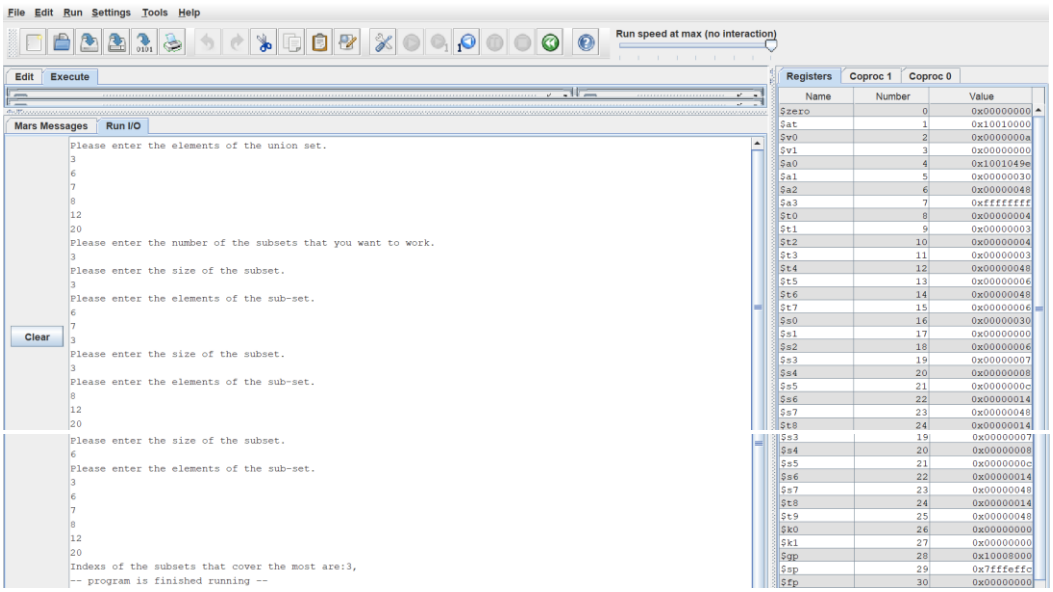
2. *Test Case 2*

In this case, elements of the UnionSet are 1,2,3,4,5,6. There are five subsets. First has 1,2. Second 3,4. Third has 5,6. Fourth has 1,2,3. Last one has 4,5,6. As you can see union of fourth and fifth subset gives us UnionSet. I can choose another combination like S1, S2, S3, but this case has the least numbers.



3. *Test Case 3*

In this case, elements of the UnionSet are 3,6,7,8,12,20 There are three subsets. First has 6,7,3. Second 8,12,20. Las one has 3,6,7,8,12,20. As you can see third subset gives us UnionSet. I can choose another combination like S1 and S2, but this case has the least numbers.



4. *Test Case 4*

In this case, elements of the UnionSet are 2,4,6,8,10,12. There are four subsets. First has 2,4. Second has 12. Third one has 6,8. Last one has 10,12 As you can see S1, S3 and S4 gives us UnionSet.