CSE 331 Computer Organization
Project 1 – MIPS Assembly
REPORT

MinSet Cover problem tries to construct a set using with minimum number of subset of that set. To achive this problem, we must find whic subset has most element than other subset that covers our set. Also we need to check that the set is empty i.e is there any remaining element or not.

In this Project, I started by reading. Opened file with syscall 13 and readed by 14. I take a buffer on memory which has 1 byte, since the reading function reads every time 1 byte character. After that code looks what the read function takes from file. Since buffer is not aligned by word property, shifting must be done. The shifted value gives a integer that correspons ascii character. If the our readed char is blank, just ,ignore it. If the readed value is new line char this means that we're going to start to read new set but i also take new line char in memory, because it indicates that end of a set or starting of new set. Every value taked on file will processed. If readed char is a number, this means we need to convert it to int. We can accomplish that by anding with 0xF. If we have a number that has more than one digit. We can achieve this by taking value shifting left and or with previous buffer. In this case we are readed integers with has any digit but not in decimal base. But this is not a problem since we compare same taked values with same readed values. Also when we reading we store the set number and where to start all sets. This is a hard coding style but 3 byte later of buffer gives us starting of sets.

After creating all sets we need a fucntion that gets our set starting address for any given set. getSet function accomplish that purpose. It iterate over sets until our search set number is zero. When this is zero, this means we found the start address of searched set.

We also need a search method that searches given element on our main list. We know our main list address and we simpliy go all the set until reach 0xa which means we come to end of set. If we have a match we simpliy return 1. At start we assume that element not in the set.

Intersection method is one of our main methods. It starts from set 1 to all the sets that using set number supplied by read function. Takes the current set, searches in main set and if it found, it increment some temporary value. After we come to the end of subset, we compare it with previous sets. End of the function, it gives us the subset that has the most intersection with main set.

Intersection gives us the most covering set. This set and intersectioned elements on main set must be deleted for remaining processes. In this case we have findAndDelete function. It simply deletes all matching elements on subset and mainset. For that purpose and to not the confuse process, we simpliy put -1 to deleted subset items and -2 for deleted main set items. With this manner we never going to match them again since they are not equal and we can find right most intersectioned subset.

We also need to check that the main set empty. This means that the main set is covered or not. It’s a simple process. We know the starting address of that. From starting address to the end(0xa), we load every element. After loading we compare it with -2. If there is a element different from -2, this implies the set isn’t empyt. Otherwise the set is empty.

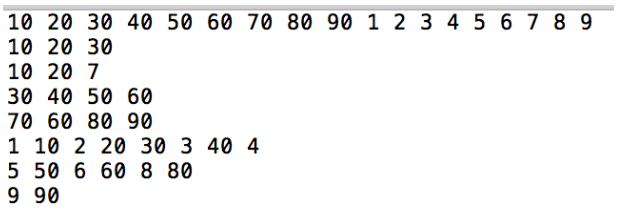
Finally we have all the tools to find min set cover. findMinSet function looks until the main set is empty. When the main set is empty, it turn back to main function. Inside loop it first get the most intersectioned set with intersection method. Since this is the set we print out that set on console. After that to clear confusion with using findAndDelete function we clear all intersectioned elements on main set and also from subset that has most intersectioned element. For new covering set, we print a blank on console to distinguish betwwen other sets. When the main set is empty, we simply return back to the main fucntion and we end program with syscall 10.

Input file must be like that:

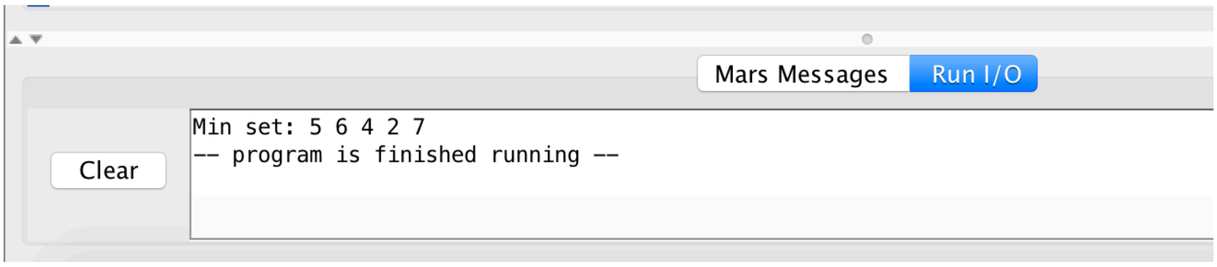
- First line represent the main set, and other represent S1, S2, S3,….etc.
- For every element on a set, there must be one space.
- After printing one set, we must put new set on new line.

RESULTS

Test1

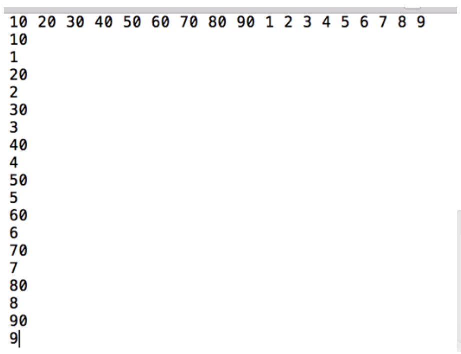


```
10 20 30 40 50 60 70 80 90 1 2 3 4 5 6 7 8 9
10 20 30
10 20 7
30 40 50 60
70 60 80 90
1 10 2 20 30 3 40 4
5 50 6 60 8 80
9 90
```

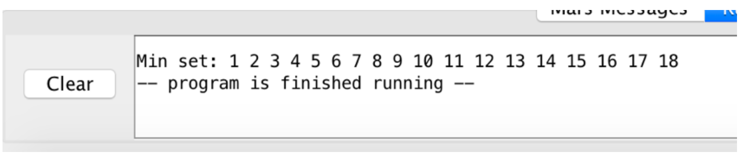


```
Min set: 5 6 4 2 7
-- program is finished running --
```

Test2



```
10 20 30 40 50 60 70 80 90 1 2 3 4 5 6 7 8 9
10
1
20
2
30
3
40
4
50
5
60
6
70
7
80
8
90
9
```



```
Min set: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
-- program is finished running --
```

Furkan Kalabalık
161044001