Hugh A. Miles II

UFID: 78686913

Section: 13A8

09/21/16

COP3530

Long Processing Time

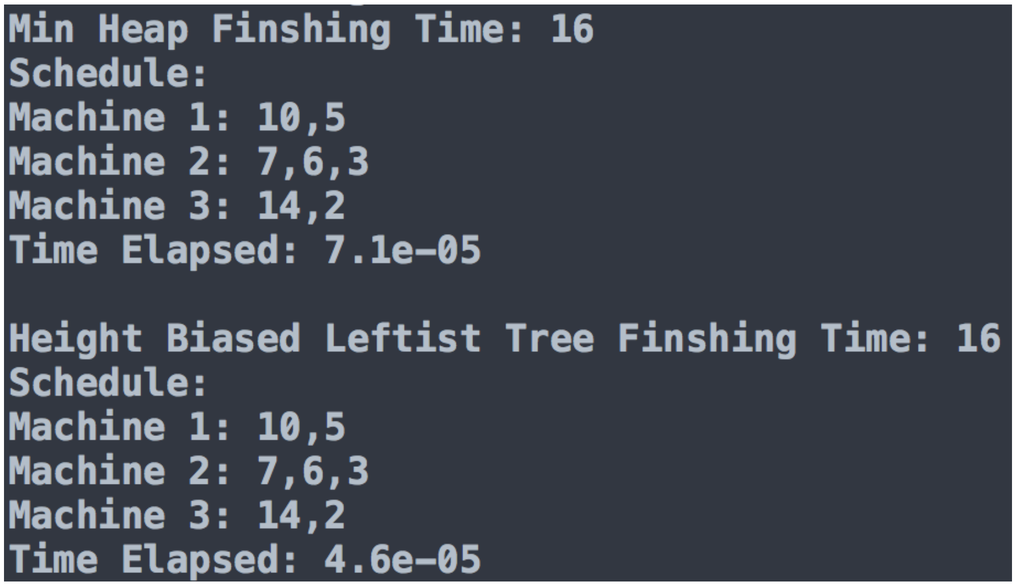
- Summary
    - My LPT program uses both a min heap and height-biased leftist tree as a priority queue of machines. The whole point of LPT is to give jobs to machines that have the smallest load (processing times sum) until no more jobs are available doing this will allow the jobs to be done in optimal time.
- Structure
    - Node<type>
        - Constructor (type initValue, int strIndex)
            - In the constructor you need to indicate the value to be set, and where the Node is being placed with the strIndex.
        - Type value
            - The data value stored in side the node.
        - Node<type>* rightChild and leftChild
            - Address for the children of this Node.

- Machine
    - Constructor(int size)
        - Gives the max number jobs that can be assigned to
    - Insert(int job)
        - This method add the job to the sum and logs
    - Print()
        - Prints the log and sum
- arrayHeap
    - push()
        - Allows you to add a machine to the heap
    - compare(int parentIndex, int childIndex)
        - checks if the parents is < then its child if not it bubbles down till it finds its position
    - realign()
        - After a pop this starts a bubble down with the last element in the heap
    - top()
        - returns the lowest machine
    - pop()
        - removes the lowest machine
    - isEmpty()
        - bool that tells you if the heap is empty
    - size()

        - gives the amount of machines in the heap
    - printTree()
        - print out the heap in Tree form
    - getMax()
        - returns the biggest machine’s sum
    - getRightChild(int parentIndex)
        - gives you right child of current index
    - getLeftChild()
        - gives you right child of current index
    - getParent(int childIndex)
        - gives parent of current childIndex
    - swap(int indexOne, int indexTwo)
        - swaps index positions in place
- minTree
    - push(type value)
        - push Machine in to heap
    - pop()
        - removes root and melds left over subtrees
    - top()
        - returns machine at the root of tree
    - print()
        - machines in Queue
    - printTree()

    - print elements of queue in tree form
- printSubTree(Node<type>* root)
    - print tree from Node given
- meld(Node<type>* currentRoot, Node<type>* toMeld)
    - takes roots of 2 trees and melds toMeld in to the current root
- checkRanks()
    - checks if the tree is leftist and recursively fixs the trees alignment
- getRank(Node<type>* n)
    - get the rank of the current Node n
- swap(Node<type>* parent)
    - swaps the parents children

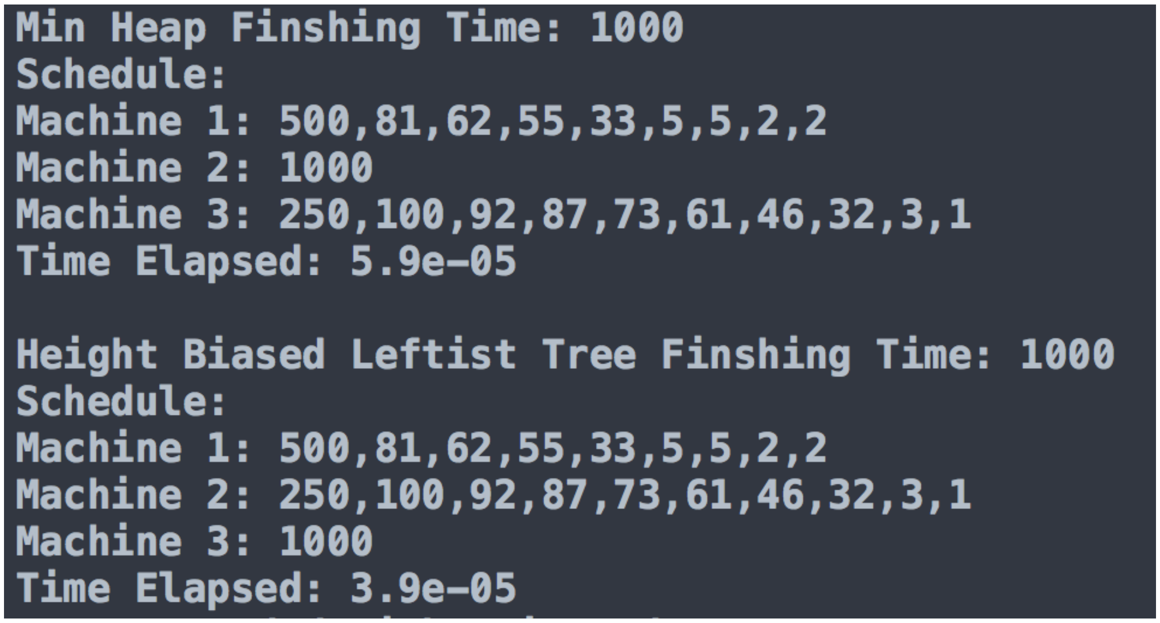
- Test Cases
    - Medium (Machine = Jobs)
        - Input
            - 3 Machines
            - 7 Jobs
                - Jobs[] = 5, 10, 14, 6, 2, 3, 7
        - Output



```
Min Heap Finshing Time: 16
Schedule:
Machine 1: 10,5
Machine 2: 7,6,3
Machine 3: 14,2
Time Elapsed: 7.1e-05

Height Biased Leftist Tree Finshing Time: 16
Schedule:
Machine 1: 10,5
Machine 2: 7,6,3
Machine 3: 14,2
Time Elapsed: 4.6e-05
```

- Small (Machines < Jobs)
  - Input
    - 3 Machines
    - 20 Jobs
      - Jobs[] = 62, 33, 100, 5, 2, 1, 1000, 61, 46, 87, 32, 81, 92, 2, 5, 3, 73, 500, 250
  - Output



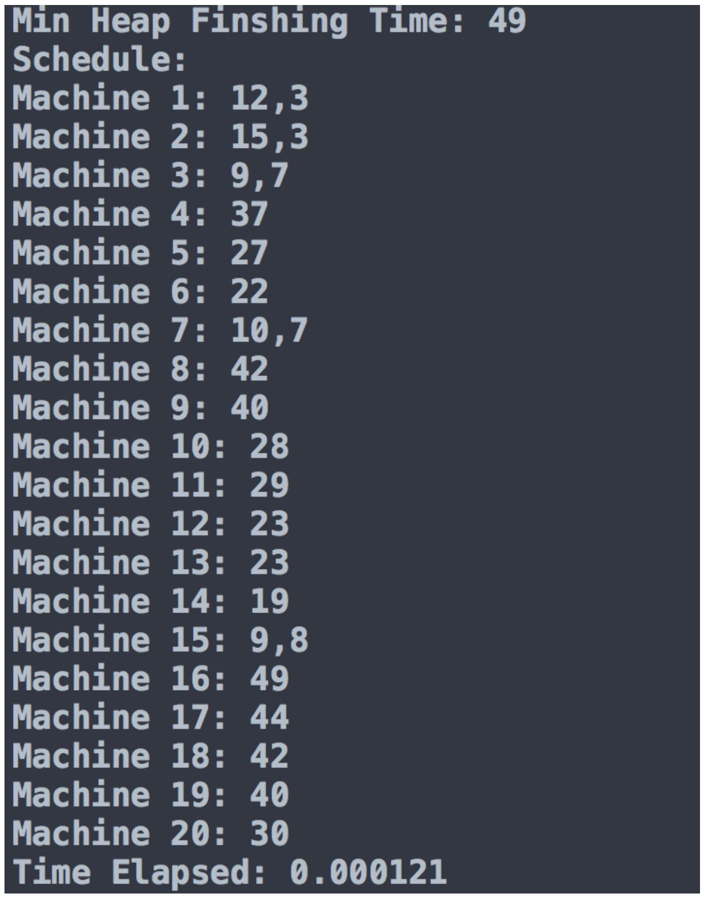
```
Min Heap Finshing Time: 1000
Schedule:
Machine 1: 500,81,62,55,33,5,5,2,2
Machine 2: 1000
Machine 3: 250,100,92,87,73,61,46,32,3,1
Time Elapsed: 5.9e-05

Height Biased Leftist Tree Finshing Time: 1000
Schedule:
Machine 1: 500,81,62,55,33,5,5,2,2
Machine 2: 250,100,92,87,73,61,46,32,3,1
Machine 3: 1000
Time Elapsed: 3.9e-05
```

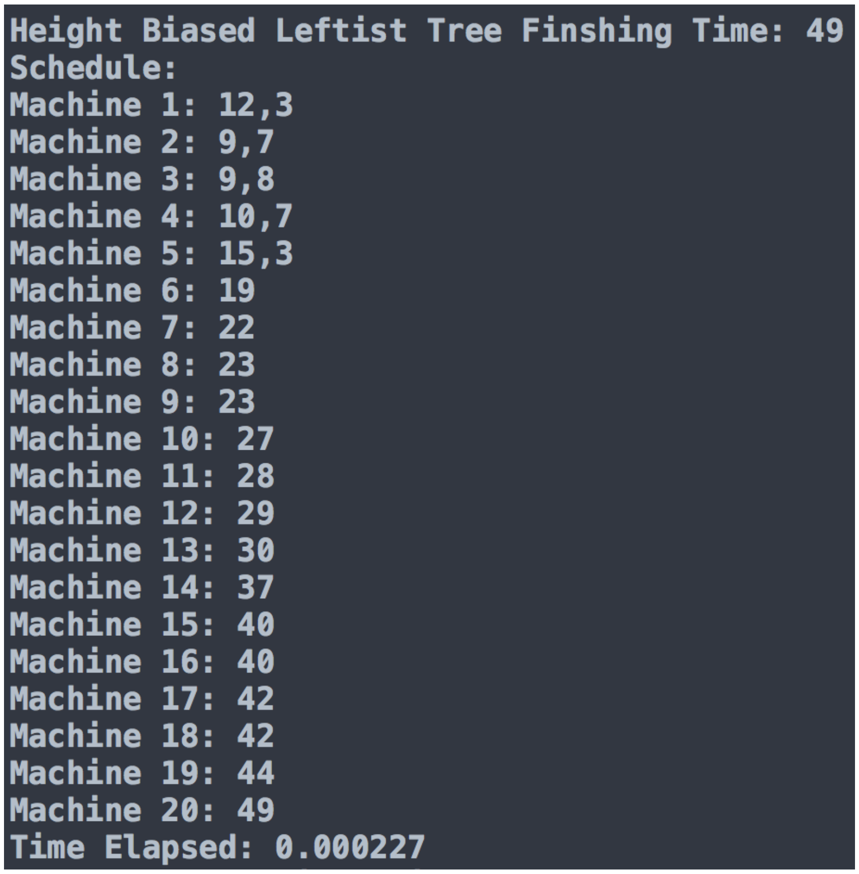
- Big (Machines > Jobs)
  - Input
    - 20 Machines
    - 25 Jobs
      - Jobs[] =

        7,49,23,8,30,22,44,28,23,9,40,14,42,42,37,3,2

        7,29,40,12,3,19,9,7,10
  - Output



```
Min Heap Finshing Time: 49
Schedule:
Machine 1: 12,3
Machine 2: 15,3
Machine 3: 9,7
Machine 4: 37
Machine 5: 27
Machine 6: 22
Machine 7: 10,7
Machine 8: 42
Machine 9: 40
Machine 10: 28
Machine 11: 29
Machine 12: 23
Machine 13: 23
Machine 14: 19
Machine 15: 9,8
Machine 16: 49
Machine 17: 44
Machine 18: 42
Machine 19: 40
Machine 20: 30
Time Elapsed: 0.000121
```



```
Height Biased Leftist Tree Finshing Time: 49
Schedule:
Machine 1: 12,3
Machine 2: 9,7
Machine 3: 9,8
Machine 4: 10,7
Machine 5: 15,3
Machine 6: 19
Machine 7: 22
Machine 8: 23
Machine 9: 23
Machine 10: 27
Machine 11: 28
Machine 12: 29
Machine 13: 30
Machine 14: 37
Machine 15: 40
Machine 16: 40
Machine 17: 42
Machine 18: 42
Machine 19: 44
Machine 20: 49
Time Elapsed: 0.000227
```