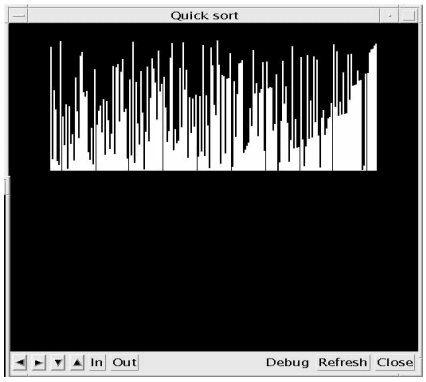
Parallel Quick Sort (Possible project task)

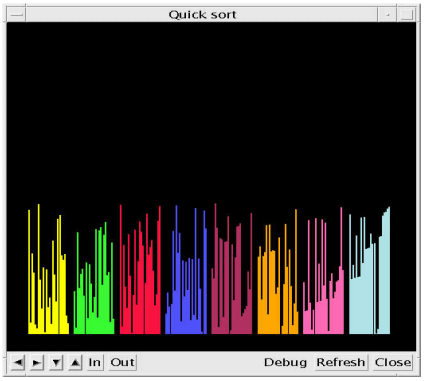
Algorithm:

1. *Divide the data into p equal parts*
2. *Sort the data locally in each processor*
3. *Perform global sort*
   - *3.1 Select pivot in each processor set*
   - *3.2 In each processor, divide the data into two sets (smaller or larger)*
   - *3.3 Split the processors into two groups and exchange data pair-wise*
   - *3.4 Merge data into a sorted list in each processor*
4. *Repeat 3.1-3.4 recursively for each processor group*

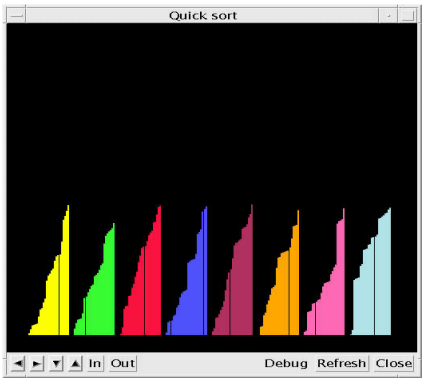
Unsorted data



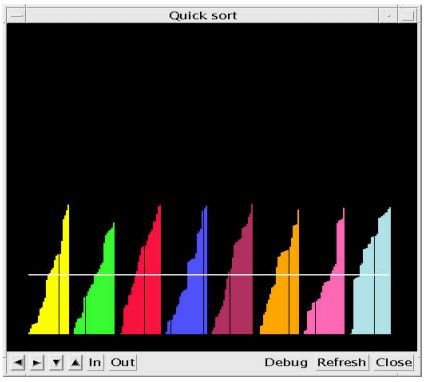
Step 1, Divide data into p equal parts



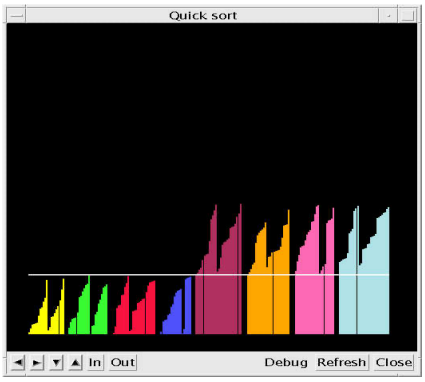
Step 2, Sort locally in each processor



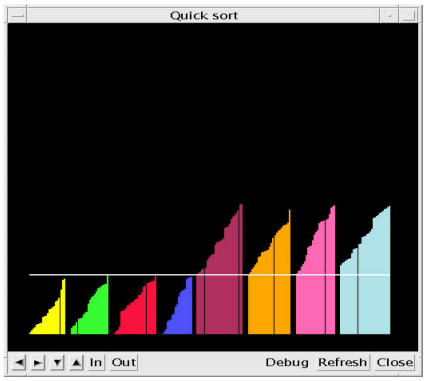
Step 3.1 Select pivot



Step 3.2, 3.3 Divide and exchange



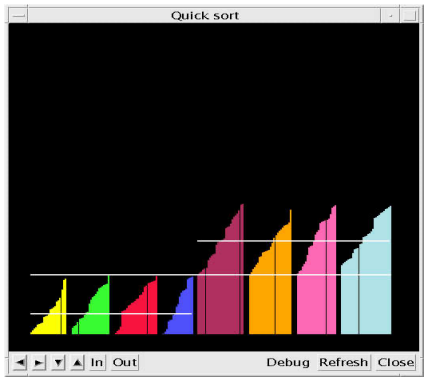
Step 3.4, Merge into a sorted list



Institutionen för informationsteknologi | www.it.uu.se | Jarmo Rantakokko

23

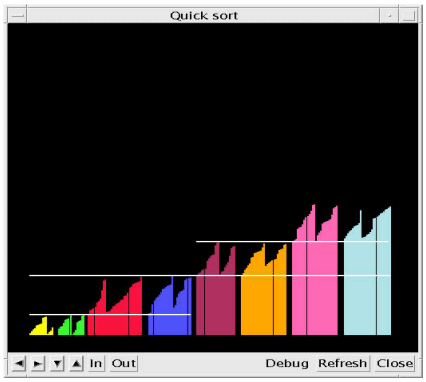
Step 3.1 Select pivot



Institutionen för informationsteknologi | www.it.uu.se | Jarmo Rantakokko

24

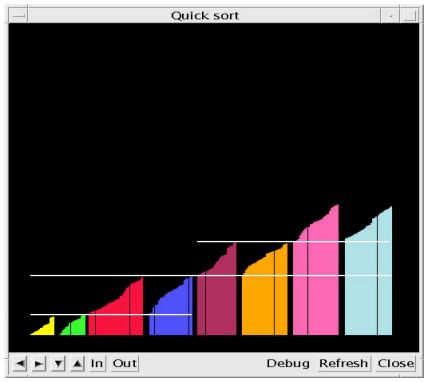
Step 3.2, 3.3 Divide and exchange



Institutionen för informationsteknologi | www.it.uu.se | Jarmo Rantakokko

25

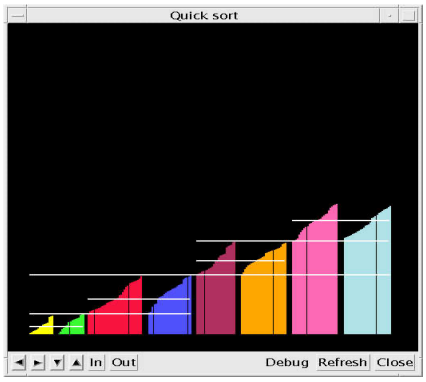
Step 3.4, Merge into a sorted list



Institutionen för informationsteknologi | www.it.uu.se | Jarmo Rantakokko

26

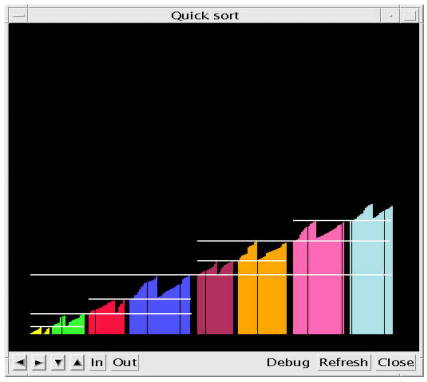
Step 3.1 Select pivot



Institutionen för informationsteknologi | www.it.uu.se | Jarmo Rantakokko

27

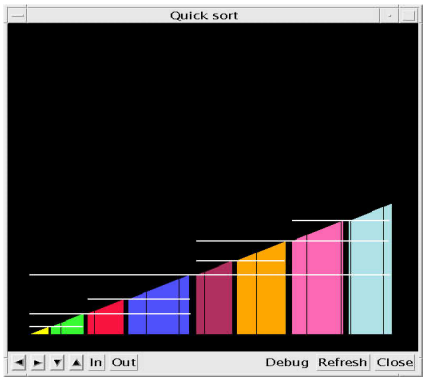
Step 3.2, 3.3 Divide and exchange



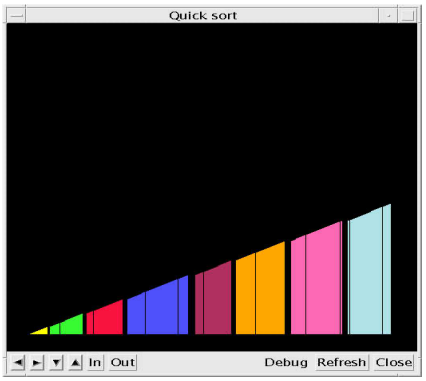
Institutionen för informationsteknologi | www.it.uu.se | Jarmo Rantakokko

28

Step 3.4, Merge into a sorted list



Final state



Pseudo code (global sort):

```
void Global_sort(data,size,myid){
    if size==1 return
    locid=myid%size
    group=myid/size

    if locid==0 pivot[group]=select(data[myid])
    Synchronize group
    splitpoint=findsplit(data[myid],pivot[group])
    wait neighbor split
    if locid<size/2
      Merge(data[myid],data[myid+size/2],lowerparts)
    else
      Merge(data[myid],data[myid-size/2],upperparts)
    wait neighbor merge
    Global_sort(data,size/2,myid)
}
```


Pivot selection strategies:

Strategy 1: Select median in processor 0 in each processor set (communicator) and step.
(OK if data equally rand, bad if almost sorted)

Strategy 2: Select the mean of all medians in respective processor set and step.
(Can give too much weight to extreme medians)

Strategy 3: Sort the medians and select the mean value of the two middlemost medians in each processor set and step.
(Independent of dist but more costly strategy)