

COP5536 Exam 1 Summer 2012

Sample Solution

1. *Add()* operation gets charged 4 and *DeleteOdds()* operation gets charged 0. When we call *Add(x)*, we use 2 immediately and store the remaining 2 with the item added to the list.

When we call *DeleteOdds()*, each element in the list has 2 credits. We use one credit from each item to pay the cost of the call to *DeleteOdds()*. Further, for every item deleted from list, we take the extra 1 credit stored at that item and place it on the next item in the list.

Thus, at the end of the call to *DeleteOdds()* all elements in the list still have 2 credits stored on them.

2. Note that we need 2 dummy runs (run length is zero) for optimal 4-way merge which must be merged first to be optimal.

4-way merge:

step 1: (0,0,100,200) = 300

step 2: (300,300,400,500) = 1500

step 3: (1500,600,700,800) = 3600

8-way merge:

(100,200,300,400,500,600,700,800) = 3600

Number of comparisons:

In 4-way scheme: For each step, loser tree initialization needs 3 comparisons (one record produced) and then each record needs 2 comparisons to output.

So, the total number of comparisons in the 4-way scheme is

$$3 + (300-1)*2 + 3 + (1500-1)*2 + 3 + (3600-1)*2 = 10803$$

In 8-way scheme: $7 + (3600-1)*3 = 10804$.

Number of disk IOs

Each merge step need 2 disk IOs: one for input and one for output. 4-way scheme needs $3*2 + 15*2 + 36*2 = 108$ and 8-way scheme $36*2 = 72$.

Suppose one comparison costs time t , then one I/O time is $100t$,

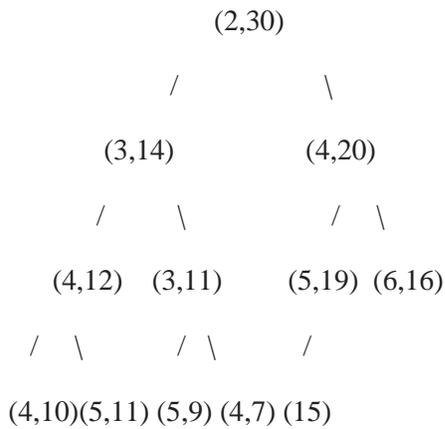
Total time cost of optimal 4-way merge scheme is: $10803t + 10800t = 21603t$

Total time cost of optimal 8-way merge scheme is: $10804t + 7200t = 18004t$

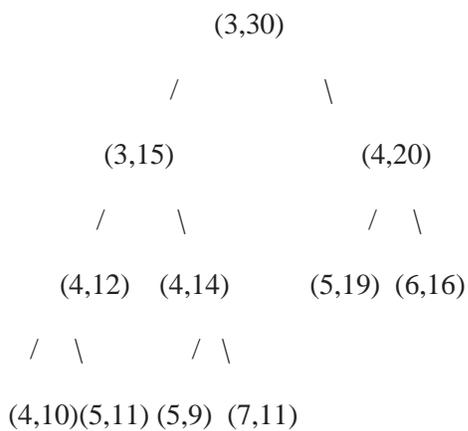
Hence, 8-way merge is better scheme than 4-way merge scheme .

3.

After insert(20)



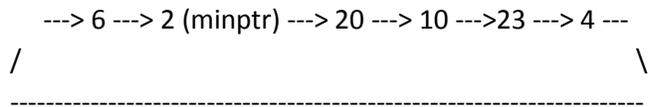
After deleteMin



Question 4.

Step 1:

Insert 6, 2, 4, 23, 10, 20



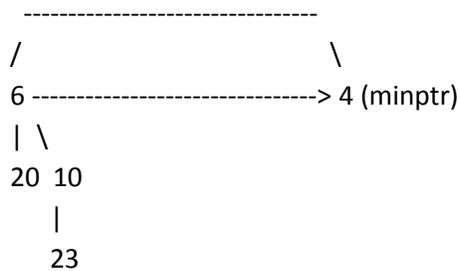
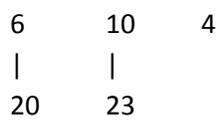
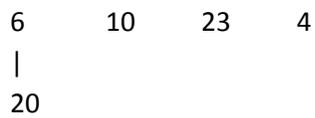
Step 2:

RemoveMin

Min node 2 removed

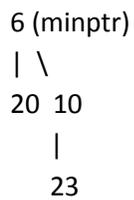
6 20 10 23 4

Pairwise combine and update the min pointer



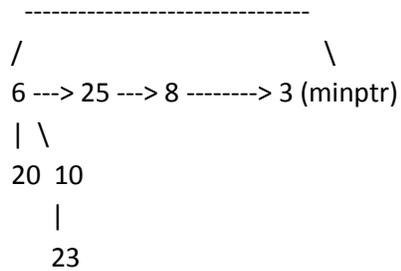
Step 3:

RemoveMin



Step 4:

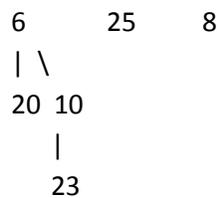
Insert 3, 8, 25



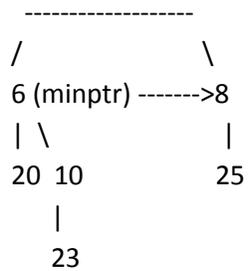
Step 5:

RemoveMin

Remove node 3



Pairwise combine and update the min pointer



Note:- There can be different solutions to this question based on the order of inserting nodes. It may also change with the technique used for inserting nodes. One thing which should not vary is the way pairwise merging is done after removing the min node. The order should be maintained after removing the min node and left to right traversal should be used for pairwise combine as shown in the lecture slides.