Sample solution for NTU AD 711R Summer 2005

1. (60) Remember that your equation should be based on the first n months not just for first 12 months. Also careful when you draw the equation.

   (a) (40) Using the aggregate method,
   First, determine an upper bound on the sum of the costs for the first n months.
   The sum of the actual monthly costs for the first n months is

$$\begin{aligned} &= 80 \times \lfloor n/12 \rfloor + 40 \times (\lfloor n/3 \rfloor - \lfloor n/12 \rfloor) + 20 \times (n - \lfloor n/3 \rfloor) \\ &= 40 \times \lfloor n/12 \rfloor + 20 \times \lfloor n/3 \rfloor + 20 \times n \\ &\leq 40 \times n/12 + 20 \times n/3 + 20 \times n \\ &= 20 \cdot (1/6 + 1/3 + 1) \\ &= 20 \cdot (3/2) \\ &= 30 \cdot n \end{aligned}$$

   (b) (20) Using the amortized cost calculated in part(a), fill in the following table.

Month	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
Actual cost	20	20	40	20	20	40	20	20	40	20	20	80
Amortized cost	30	30	30	30	30	30	30	30	30	30	30	30
Potential()	10	20	10	20	30	20	30	40	30	40	50	0

Table 1: Maintenance Contract

2. (60) S=180 records
   n=1000 records
   m runs
   $t_s = 6$ ms
   $t_l = 4$ ms
   $t_t = 0.1$ ms / record

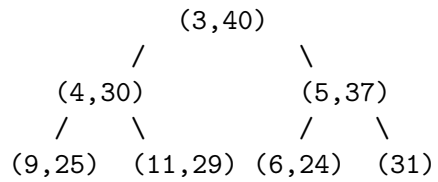
   (a) $b = \lfloor S/(2k+2) \rfloor = \lfloor 180/(2*8+2) \rfloor = 180/18 = 10$
   (b) time to read a buffer = (6 + 4 + (10)*0.1) = (10 + (10)*0.1) ms = 11 ms
   (c) number of buffers per pass = $\lceil n/b \rceil = \lceil 1000/10 \rceil = 100$
   (d) input time per pass = $(b) * (c) = 11 * 100 = 1100ms$

(e) number of passes = $\lceil \log_k m \rceil = \log_8 m$

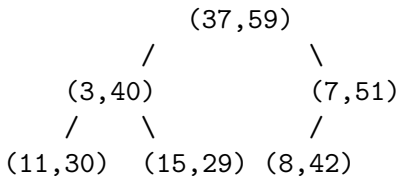
(f) total input time = $(d) * (e) = 1100 * \log_8 m$

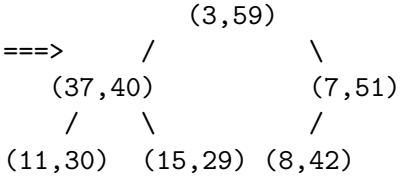
3. (40)

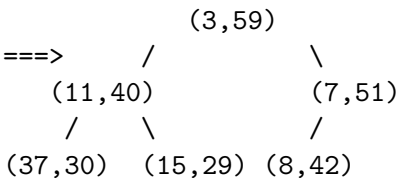
(a) (15)

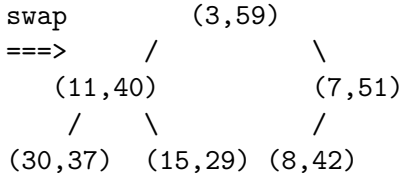


(b) (25) $RemoveMin()$
Remove 2 from the root.
Remove 37 from the last node and insert it into the root.



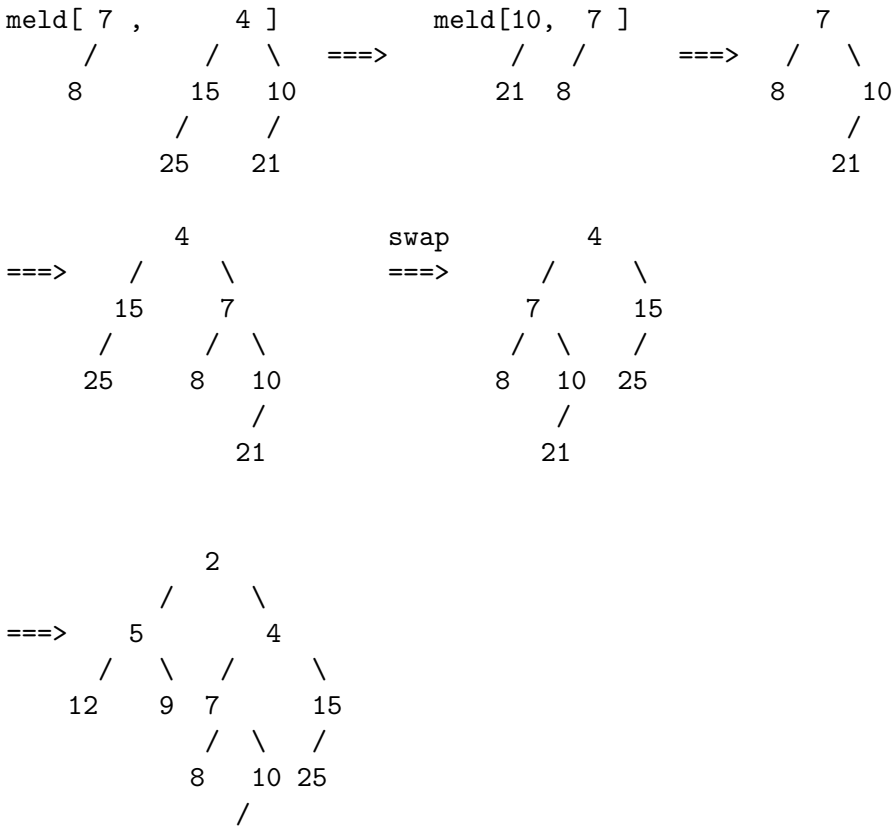






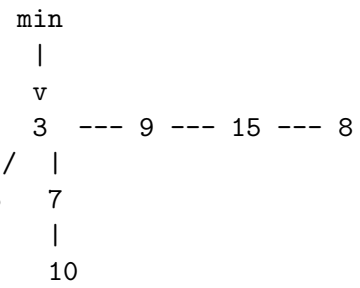
4. (30) Remember the definition of the height-biased min leftist tree.
   - has min-heap property
   - swap only if needed

   Meld right subtree with smaller root and all of the other tree.

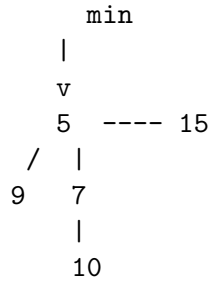


5. (60)

   (a) (10)



   (b) (20) Pairwise combine after deletion



   (c) (30) create and initial tree table = $O(MaxDegree)$.
   Examine t min trees and pairwise conbime = $O(t)$
   Collect remaining trees from tree table, reset table entire to null = $O(MaxDegree)$
   Thus, overall complexity of Remove min = $O(MaxDegree + t)$