University of Nevada, Las Vegas Computer Science 477/677 Fall 2021

Answers to Assignment 4: Due Monday October 4, 2021 11:59 pm

1. (a) Name three kinds of priority queue.

   stack, queue, heap.

   (b) The items in a priority queue always represent unfulfilled obligations

2. Give an asymptotic solution for the recurrence

   $F(n) = 2F(n-1) + 3F(n-2)$

   We assume that $F(n)$ is an increasing function of n. F is clearly exponential. Assume that there are constants A, B such that $F(n) = AB^n$. We then have

$$\begin{aligned} AB^n &= 2AB^{n-1} + 3AB^{n-2} \\ \text{Divide both sides by } AB^{n-2}: & \\ B^2 &= 2B + 3 \\ \text{By the quadratic formula}: & \\ B &= \frac{2 \pm \sqrt{16}}{2} \\ &= 1 \pm 2 \end{aligned}$$

   B is non-negative since F(n) is increasing. Thus

$$\begin{aligned} B &= 3 \\ F(n) &= 3^n A \\ &= \Theta(3^n) \end{aligned}$$

3. Work problem 2.20 on page 74 of your textbook. Let $X[1 \ldots n]$ be an array of integers and let $M = \max_i X_i - \min_i X_i$. Find an $O(n + M)$ time algorithm for sorting X. For small M, this is linear time: why doesn't the $\Omega(n \log n)$ lower bound apply in this case?

   1. Let $A = \min X_i$.
   2. Declare array of integers $K[M+1]$, initialize to all zeros.
   3. For all i from 1 to n, increment $K[x_i - A]$ by one.
   4. For each j from 0 to M, output $K[j]$ copies of $j + A$

```
for(int i = 1; i <= n; i++)
 K[X[i]+A]++;
for(int j = 0; j <= M; j++)
 for(int k = 0; k < K[j]; k++)
  cout << j+A << endl;
```

   The orginal values of $\{x_i\}$ will be written in sorted order.

   The algorithm does not use the comparison model of computation, therefore the $\Omega(n \log n)$ lower bound does not hold.

4. Walk though heapsort with input array UBRYPQSVFMTX by filling in the array below and on the next page. Warning: the number of blank rows is not correct.

1	2	3	4	5	6	7	8	9	10	11	12	
U	B	R	Y	P	Q	S	V	F	M	T	X	
U	B	R	Y	P	X	S	V	F	M	T	Q	bubbledown Q
U	B	R	Y	T	X	S	V	F	M	P	Q	bubbledown P
U	B	X	Y	T	R	S	V	F	M	P	Q	bubbledown R
U	Y	X	B	T	R	S	V	F	M	P	Q	bubbledown B
U	Y	X	V	T	R	S	B	F	M	P	Q	bubbledown B
Y	U	X	V	T	R	S	B	F	M	P	Q	bubbledown U
Y	V	X	U	T	R	S	B	F	M	P	Q	bubbledown U heapify completed
Q	V	X	U	T	R	S	B	F	M	P	Y	Y in final position
X	V	Q	U	T	R	S	B	F	M	P	Y	bubbledown Q
X	V	S	U	T	R	Q	B	F	M	P	Y	bubbledown Q heap order restored
P	V	S	U	T	R	Q	B	F	M	X	Y	X in final position
V	P	S	U	T	R	Q	B	F	M	X	Y	bubbledown P
V	U	S	P	T	R	Q	B	F	M	X	Y	bubbledown P heap order restored
M	U	S	P	T	R	Q	B	F	V	X	Y	V in final position
U	M	S	P	T	R	Q	B	F	V	X	Y	bubbledown M
U	T	S	P	M	R	Q	B	F	V	X	Y	bubbledown M heap order restored
F	T	S	P	M	R	Q	B	U	V	X	Y	U in final position
T	F	S	P	M	R	Q	B	U	V	X	Y	bubbledown F
T	P	S	F	M	R	Q	B	U	V	X	Y	bubbledown F heap order restored
B	P	S	F	M	R	Q	T	U	V	X	Y	T in final position
S	P	B	F	M	R	Q	T	U	V	X	Y	bubbledown B
S	P	R	F	M	B	Q	T	U	V	X	Y	bubbledown B heap order restored
Q	P	R	F	M	B	S	T	U	V	X	Y	S in final position
R	P	Q	F	M	B	S	T	U	V	X	Y	bubbledown Q heap order restored
B	P	Q	F	M	R	S	T	U	V	X	Y	R in final position
Q	P	B	F	M	R	S	T	U	V	X	Y	bubbledown B heap order restored
M	P	B	F	Q	R	S	T	U	V	X	Y	Q in final position
P	M	B	F	Q	R	S	T	U	V	X	Y	bubbledown M heap order restored
F	M	B	P	Q	R	S	T	U	V	X	Y	P in final position
M	F	B	P	Q	R	S	T	U	V	X	Y	bubbledown F heap order restored
B	F	M	P	Q	R	S	T	U	V	X	Y	M in final position
F	B	M	P	Q	R	S	T	U	V	X	Y	bubbledown B heap order restored
B	F	M	P	Q	R	S	T	U	V	X	Y	F in final position. Sorted.

5. Walk through the steps of mergesort to sort the file YUEPGJQTBHZRWCNK.

   YUEPGJQTBHZRWCNK
   YUEPGJQT BHZRWCNK
   YUEP GJQT BHZR WCNK
   YU EP GJ QT BH ZR WC NK
   UY EP GJ QT BH RZ CW KN
   EPUY GJQT BHRZ CKNW
   EGJPQTUY BCKHNRWZ
   BCEGJKHPNQRTUWYZ

6. Walk through the steps of polyphase mergesort to sort the file YUEPGJQTBHZRWCNK. **Remember!** Polyphase mergesort is not the same as mergesort.

   Y U EP GJQT BHZ RW CN K.

   Y EP BHZ CN
   U GJQT RW K

   UY BHRWZ
   EGJPQT CKN

   EGJPQTUY
   BCHKNRWZ

   BCEGHJKNPQRTUWYZ