University of Nevada, Las Vegas Computer Science 477/677 Fall 2024

Anssers to Assignment 3:

This assignment will not be graded. Answers will be posted on Tuesday September 23, 2024.

1. Here is a prefix code for the alphabet {E,E,H,L,O,R,W}.

D	00
E	10
H	110
L	010
O	011
R	1110
W	1111

Why is it called a prefix code?
Because no code string is a prefix of any other, making decoding unambiguous.

Decode the string
1 1 0 1 0 0 1 0 0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 0 1 0 0 0
H E L L O W O R L D

3. Solve the following recurrences using the anti-derivative method.

   (a) $F(n) = F(n-1) + n^2$

   $$\frac{F(n) - F(n-1)}{1} = n^2 \; F(n) = \Theta(n^3)$$
   $$F'(n) = n^2$$
   $$F(n) = \Theta(n^3)$$

   (b) $F(n) = F(n - \log n) + \log n$

   $$\frac{F(n) - (F(n) - \log n)}{\log n} = \frac{\log n}{\log n}$$
   $$F'(n) = 1$$
   $$F(n) = \Theta(n)$$

4. Solve the following recurrences using the master theorem.

   (a) $F(n) = 2F(n/2) + n$

   A = 2, B = 2, C = 1

   $F(n) = \Theta(n \log n)$

   (b) $F(n) = F(n/2) + 1$

   A = 1, B = 2, C = 0

   $F(n) = \Theta(\log n)$

   (c) $F(n) = 4F(n/2) + n$

   A = 4, B = 2, C = 1, $\log_B A = 2 > C$

   $F(n) = \Theta(n^2)$

(d) $F(n) = 4F(n/2) + n^2$

A = 4, B = 2, C = 2, $\log_B A = 2 = C$

$F(n) = \Theta(n^2 \log n)$

(e) $F(n) = 4F(n/2) + n^3$

A = 4, B = 2, C = 3, $\log_B A = 2 < C$

$F(n) = \Theta(n^3)$

5. Solve the following recurrences.

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

$F(n) = \Theta(2^n)$

(b) $F(n) = F(\sqrt{n}) + 1$ Hint: use substitution.

Let $m = \log n$, and $F(n) = G(\log n) = G(m)$. Thus, $F(\sqrt{n}) = G(\log \sqrt{n}) = G(\frac{\log n}{2}) = G(\frac{m}{2})$
Substituting, we have the recurrence

$G(m) = G(m/2) + 1$

By the Master theorem we have $G(m) = \Theta(\log m)$ Thus $F(n) = \Theta(\log \log n)$.

(c) $F(n) = F(\log n) + 1$ You may have a hard time finding this on the internet.

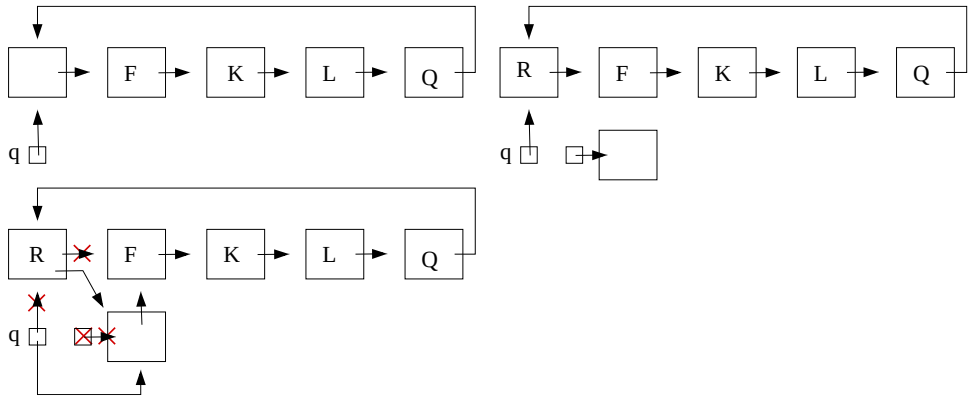
$F(n) = \Theta(\log^* n)$

6. Find the value of each expression.

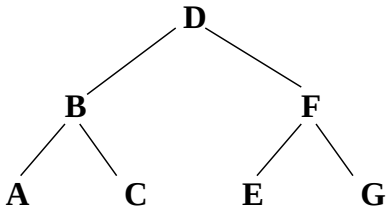
(a) $\log_4 \sqrt{2} = \frac{1}{4}$

(b) $4^{\log_9 27} = 4^{\frac{3}{2}} = 8$

7. Illustrate a circular queue whose items are F, K, L, Q in that order. Then, illustrate the steps of inserting item R.



8. Given the following binary tree:



Write the nodes in inorder, preorder, postorder, and level order.

(a) Inorder: ABCDEFG

(b) Preorder: DBACFEG

(c) Postorder: ACBEGFD

(d) Level order: DBFACEG

9. Suppose a stack of integers is implemented as a linked list, as in the handout:

```
struct stacknode;
typedef stacknode*stack;
struct stacknode
 {
  int item;
  stack link;
 }
```

Write C++ code for the operators push and pop.

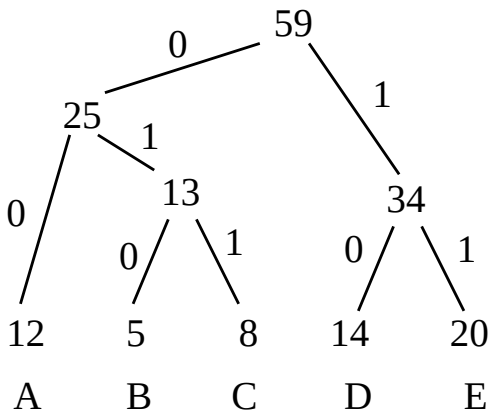
```
void push(stack&s,int newitem)
 {
  stack x = new stacknode;
  x->item - newitem;
  x->link = s;
  s = x;
 }

int pop(stack&s)
  // input condition: s is not empty
 {
  int rslt = s->item;
  s = s->link;
  return rslt;
  // memory leak is ignored
 }
```

10. Assume that A is an array declared as follows: `int A[N];` Finish writing C++ code for determining whether a given number x is an entry of A, assuming that A is sorted in increasing order.

```
bool findinA(int x)
 {
  int lo = 0; // x is not A[i] for any i < lo
  int hi = N; // x is not A[i] for any i >= N
  bool rslt = false;
  while (lo <= hi and not rslt)
   {
    int mid = (lo+hi)/2;
    if(A[mid] < x) hi = mid;
    else if(A[mid] > x) lo = mid+1;
    else rslt = true;
   }
  return rslt;
 }
```

11. Use Huffman's algorithm to find an optimal prefix code for the alphabet $\{A, B, C, D, E\}$ with the following frequencies.

A	12	00
B	5	010
C	8	011
D	14	10
E	20	11



12. Given the list E U W R B Q P X H L Z G, walk through two sorting algorithms for that list, polyphase mergesort and heapsort.

Polyphase mergesort
EUW R BQ PX HLZ G
EUW BQ HLZ
R PX G
ERUW GHLZ
BPQX
BEPQRUWX
GHLZ
BEGHLPQRUWXZ

1	2	3	4	5	6	7	8	9	10	11	12	
E	U	W	R	B	Q	P	X	H	L	Z	G	
E	U	W	R	Z	Q	P	X	H	L	B	G	bubbledow B
E	U	W	X	Z	Q	P	R	H	L	B	G	bubbledown R
E	Z	W	X	U	Q	P	R	H	L	B	G	bubbledown U
Z	E	W	X	U	Q	P	R	H	L	B	G	bubbledown E
Z	X	W	E	U	Q	P	R	H	L	B	G	bubbledown E
Z	X	W	R	U	Q	P	E	H	L	B	G	b E heapify completed
G	X	W	R	U	Q	P	E	H	L	B	**Z**	swap Z G
X	G	W	R	U	Q	P	E	H	L	B	**Z**	bubbledown G
X	U	W	R	G	Q	P	E	H	L	B	**Z**	bubbledown G
X	U	W	R	L	Q	P	E	H	G	B	**Z**	b G heap order restored
B	U	W	R	L	Q	P	E	H	G	**X**	**Z**	swap X B
W	U	B	R	L	Q	P	E	H	G	**X**	**Z**	bubbledown B
W	U	Q	R	L	B	P	E	H	G	**X**	**Z**	b B heap order restored
G	U	Q	R	L	B	P	E	H	**W**	**X**	**Z**	swap W G
U	G	Q	R	L	B	P	E	H	**W**	**X**	**Z**	bubbledown G
U	R	Q	G	L	B	P	E	H	**W**	**X**	**Z**	bubbledown G
U	R	Q	H	L	B	P	E	G	**W**	**X**	**Z**	b G heap order restored
G	R	Q	H	L	B	P	E	**U**	**W**	**X**	**Z**	swap U G
R	G	Q	H	L	B	P	E	**U**	**W**	**X**	**Z**	bubbledown G
R	L	Q	H	G	B	P	E	**U**	**W**	**X**	**Z**	b G heap order restored
E	L	Q	H	G	B	P	**R**	**U**	**W**	**X**	**Z**	swap R E
Q	L	E	H	G	B	P	**R**	**U**	**W**	**X**	**Z**	bubbledown E
Q	L	P	H	G	B	E	**R**	**U**	**W**	**X**	**Z**	b E heap order restored
E	L	P	H	G	B	**Q**	**R**	**U**	**W**	**X**	**Z**	swap Q E
P	L	E	H	G	B	**Q**	**R**	**U**	**W**	**X**	**Z**	b E heap order restored
B	L	E	H	G	**P**	**Q**	**R**	**U**	**W**	**X**	**Z**	swap P B
L	B	E	H	G	**P**	**Q**	**R**	**U**	**W**	**X**	**Z**	bubbledown B
L	H	E	B	G	**P**	**Q**	**R**	**U**	**W**	**X**	**Z**	b B heap order restored
G	H	E	B	**L**	**P**	**Q**	**R**	**U**	**W**	**X**	**Z**	swap L G
H	G	E	B	**L**	**P**	**Q**	**R**	**U**	**W**	**X**	**Z**	b G heap order restored
B	G	E	**H**	**L**	**P**	**Q**	**R**	**U**	**W**	**X**	**Z**	swap H B
G	B	E	**H**	**L**	**P**	**Q**	**R**	**U**	**W**	**X**	**Z**	b B heap order restored
E	B	**G**	**H**	**L**	**P**	**Q**	**R**	**U**	**W**	**X**	**Z**	swap G E
B	**E**	**G**	**H**	**L**	**P**	**Q**	**R**	**U**	**W**	**X**	**Z**	swap E B