PASCAL 2 Study Guide (PT 1)

Ch 2 (all) Stacks Suggested Probs already given

Ch 3 (recursion) p105 1 or 2, 7 p120 look at 1, 2, 5 p133 5 look at 6
p154 3 (iterative method) p.157 1

Ch 4 (link lists & queues) [OMIT 4.3 on Simulation] p166 look at 9, 10
p188 4 (algorithms) p215 look at 10

Ch 5 (Pascal Pointers) p.228 3 look at 10 [OMIT THE EXAMPLES in 5.2, 5.3]

Ch 6 (Trees) [JUST 6.1 & 6.2 but not threads] p261 1–7; 6 & 7 by recursion
p274 look at 7

§9.2 (BST, AVL) {pp438-445; 447-456} p462 2, 9 look at 4, 8

OTHER PROBLEMS

1. Give the LIST or LISTS (using pictures with (1) (info, next) nodes and the value of AVAIL corresponding to the "data box" array at right.

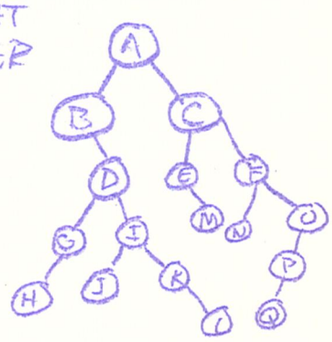
	info	next
1	S	6
2	Y	0
3	A	1
4	W	2
5		10
6	T	0
7		9
8	X	3
9		0
10		7

2. Build the BST (AVL) for the data

   50, 60, 70, 20, 30, 40, 80, 90, 100

3. Write an algorithm which will
   A. merge two ordered lists keeping copies
   B. merge two ordered lists keeping only one copy of duplicates
   C. Deleting every 2nd element of a list
   D. Reverses the list after the tenth element

4. LIST THE NODES OF THE TREE TO RIGHT IN A. PREORDER B. POSTORDER C INORDER



5. Write A*(B+C*D)-E in postfix

6. Procedure Enter (Var P NodePointer)

```
Read(char)
if Char = '*' then p ← NIL
ELSE | new(p)
     | info(p) ← char
     | Enter (right(p))
     | Enter (left(p))
END ELSE
```

INPUT SEQUENCE
AL*Z**I W**B**
What is the resulting structure? (use (T) pictures.)

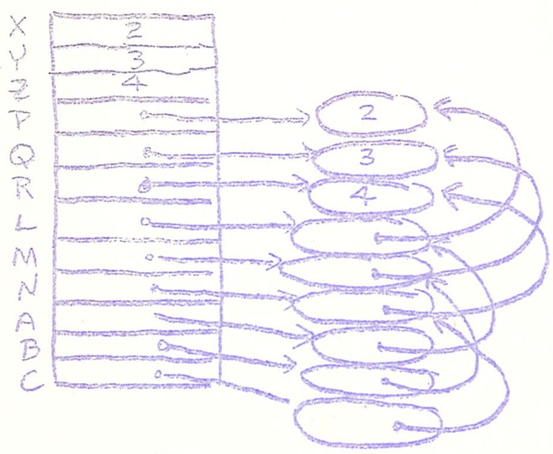
7. Procedure ReverseThem
   var A char
   if not eoln then read(A)
   ReverseThem
   Write(A)

use your knowledge of the run-time stack to write a non recursive routine. This routine should imitate what is being done by run-time stack. USE stack Primitives.

8. Write a recursive procedure which list the nodes of a binary tree in-the order A. reverse preorder B. by level and within a level from left to right (Breath First Search) (Hint keep two Queues

9. Given X, Y, Z : integers P, Q, R : ↑integer L, M, N : ↑↑integer
A, B, C : ↑↑↑integer

X ← 2, Y ← 3, Z ← 4, NEW(P, Q, R, L, M, N, A, B, C), P↑ ← X, Q↑ ← Y, R↑ ← Z
L↑ ← P, M↑ ← Q, N↑ ← R, A↑ ← L, B↑ ← M, C↑ ← N.



yields the picture to the left

Those things in the box are global variables
Ovals are in "the Heap"
Obviously this picture could be be drawn a little clearer

END GIVEN

Now we do: M↑↑ ← 5, P ← Q, NewA, A↑ ← N, P↑ ← 6, B↑↑ ← N↑, Z ← 7, L↑↑ ← P↑ + B↑↑↑, dispose(M), C ← A, A↑ ← N

Draw The Picture.

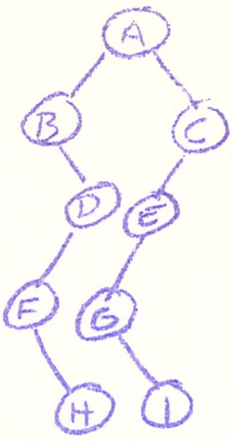
10. Prove the inorder Precessor of a node pointed to by P can be found by taking ~~left~~ right sons of ~~left~~ right(P) til we we come to a node with a nil leftson. ~~How do we modify it~~ (Assume p has two children)

11. List the primitives (and sort of what they do) for
1. STACK
2. QUEQUE
3. ~~LIST~~ LIST
4. BST.

P2 TPT2.

1. Files: A. Write a Pascal Procedure which will copy file F to File G without using the operations read or write.
   B. If P points to a file what is P^^? CAN YOU PASS P AS A VALUE PARAMETER?
   C. How many comparisons ~~is~~ at ~~how~~ most is needed to merge two sorted files one with m elements the other with n elements
2. Random Numbers A. Write a function which returns 1 with Prob. 1/4, 2 with Prob. 1/3 and 3 with Prob 5/12. (Assume Random)
   B. Write a function which returns a random integer i with $2 \le i \le 12$
3. Graphs Given any one of the following find the other two
   A. Picture of the graph B. Adjacency Matrix C, Adjacency List(s)
4. Simulation A. Consider the following list of pairs (Arrival Time, Service Times): (3,2) (6,1) (8,5) (10,3) (12,4) (20,5) (22,4) (26,3) Find the maximal and average time in Queue for (i) single server, single channel [Also give an event log] and for (ii) ~~three~~ 2 servers, single channel
   B. The good doctor actually has 3 Queues one for students, one for TA's and one for Prof.'s with the obvious priorities Do an event log if Students (10,4) (15,10) (20,7) TA's (13,2) (14,3) (22,2) Prof (16,1). Again pairs are (arrival time, service time). What is the average length of the sum of lengths of the queues,
   (Other "Stat"'s: Average System time; % use of server(s) Average interarrival time, average ~~system~~ service time
5. Sorting A. General (See 1C) (i) What are the five broad classes of sorting algorithms. (ii) SORT 1 takes $10n\log_2 n$ time, Sort 2 takes $\frac{1}{3}n^2$ time, SORT 3 takes 20n time for what values of n is SORT 1 fastest? SORT 2? SORT 3?
   (iii) How many comparisons does the following make
   For i ← 1 to n-1 do For j ← 1 to n-i do if $x_j > x_{j-1}$ then $x_j \leftrightarrow x_{j+1}$
   How much space does it use. What is its name? How much space does in use? Is it stable? How many interchanges does it make if X = (9, 8, 7, 1, 2)
   (iv) Design and explain a new method for Sorting (in English) you have ∞ memory and ∞ time.
   (v) Show N-1 comparisons is necessary and sufficient to find the largest element of an array with N elements
   (vi) Show $N + \log_2 N - 2$ Comparisons is sufficient to find both the largest and 2nd largest elements of an array with N elements [Hint divide in half]
   B. Radix Sort: Show the Queue after One pass of radix sort if the Starting Que is: TOM | KAT | ACE | ABC | AAZ | DOT | FOR | KAT | XYZ | ZEA | HEL | HIT | HUH | CHT | SEC | SEX | LOW | OFF

C. HEAPSORT (i) For the tree to right write the level order number next to each node, (ii) How many comparisons? How much space? (iii) If the initial Array is (1,2,3,4,5,6,7,8,9) What is the array after the 1ST Heap is created? After the Next Heap is formed? (iv) What does this have to do with the run-time Heap? (v) Write the "subroutine" restore both recursively and non-recursively. (vi) Is it stable?



D. QUICKSORT (i) Comparisons? Space? stable? (ii) Why is it correctly named? Why is it incorrectly named? (iii) If the test value is the first element of the array and the array is (57168 2394) what is the result after REARRANGE? What is the return value? (lb=1 ub=9) (iv) What if lb=4 ub=8? (V) Removing Recursion: using a Stack S ~~with~~ remove the recursion ~~for~~ from PostOrder transversal of a binary tree. ~~that~~ make the stack hold records NodePointer, Flag where Flag indicates if it is the 1ST or 2nd time on the stack.

E. Shellsort (i) If the array is (5 7 1 6 8 2 3 9 4) and the first increment is 5 what is the resulting array after the "1st Pass" or "increment" (ii) Pg 413 #4. (iii) What is insertion sort? Comparisons? what is it good for?

F. Type FilePointer = ↑File of integer; ~~Node=Record Data: FilePointer; left, right: NodePointer; End;~~

VAR X : Array [0..m-1] of integer $m = 2^n$

TREE: Array [1.. 2m-1] of FilePointer rewrite(Tree[i]↑)

For i = m to 2m-1 do New(Tree[i]), Write(Tree[i]↑, X[i-m])

For i = m-1 downto 1 do reset(Tree[2i]↑ & Tree[2i+1]↑)

(new(tree[i]), rewrite(tree[i]↑) tree[i]↑ ← merge tree[2i]↑ & Tree[2i+1]↑)

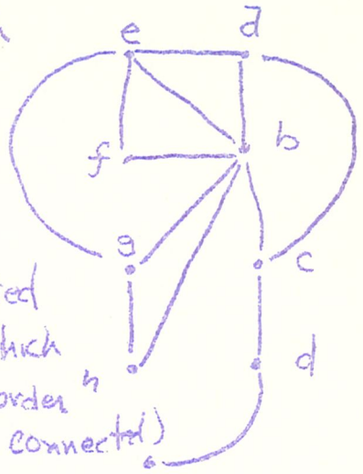
(i) Write Merge (watch out there is no ∞)

(ii) How Many Comparisons? How much space? ~~Decrease the space to 2m-1~~ Improve the space requirements

(iii) In English what is going on?

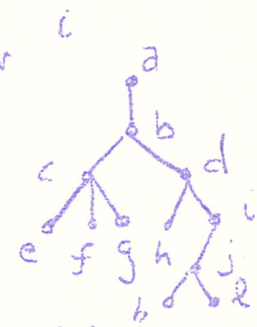
P2PT3

1. Find spanning tree (draw them) for the graph to the right by A. DFS B. BFS (note there is exactly one correct answer for each)

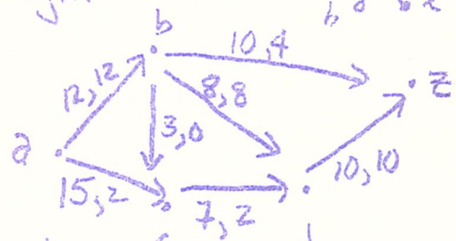


2. Assume the nodes of a graph are the numbers 1..n. The graph and the edges are stored in an adjacency matrix ADJ of boolean. Visited is a set of 1..n. Write a Pascal Routine which will visit these nodes in A. DFS order, B. in BFS order C. DFS without recursion. (Assume the graph is connected) (Be sure to show the initial routine call.)

3. For the tree to right. List the nodes in pre order in post order. Draw the corresponding binary tree and then list its nodes in pre order, in order and post order. Which are the same and why?



4. For the network flow graph to the right.
   A. "Label" as many nodes as possible
   B. Show the edges and the changes that result when we use these labels to increase the flow



5. For the same graph show the order in which Prims Algorithm would pick the edges (use the node a as a starting point and the first number as the cost of the edge) List the order of edges as Kruskal Algorithm would pick them.

6. Build a 2-3 tree for the data: 10, 15, 95, 55, 50, 60, 35, 45, 80, 20, 5, 75, 90 Show the tree after every insert.

7. In the text p387 is the code for straight selection sort. This code has two loops and inner one on j and an outer one on i. Find Assertions for these loops.

8. There are 800 students keyed by Soc. Sec No to be hashed into a table with 1000 places (0..999) Consider the following functions

   F(s) = s mod 1000 G(s) = (s DIV 1000000) H(s) = sum of the digits

   Assume these are FSU students. Which of these functions in S will be good Hash functions. (A) None (B) F only (C) H only (D) F and H only (E) G and H only.

9. Here are a sequence of INSERT, FIND. and DELETES. Answer each of the FINDS true or false. Show the hash table at the end of the sequence. Here is the Hash table at the beginning. [Create your own problem]

10. Explain chaining for a hash table. Linear probing. rehashing.

11. The good doctor has a metal file with 27 draws each has a letter on it. (except one is blank). The file is keyed to Author. What data structure is this? All particulars!

12. How many comparisons in a binary ~~seee~~ search of 1 2 3 4 5 6 7 8 9 do you make to find 1? 3? How about a sequential search for 3? 1?

13. Define a B-tree of order 5.

ANSWERS:

1 A. ABDHEICFJKG

B. HDIEBJKFGCA

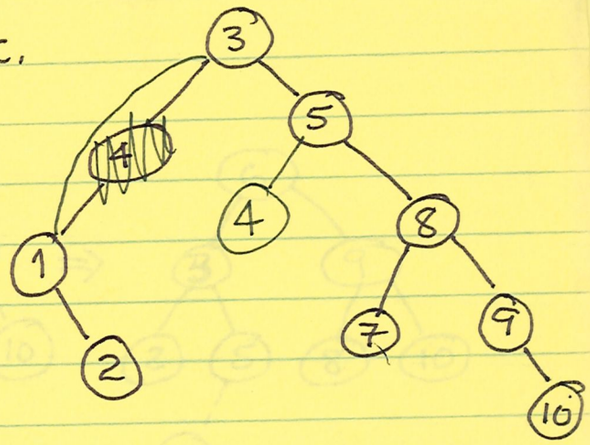
C. DHBIEAJFKCG

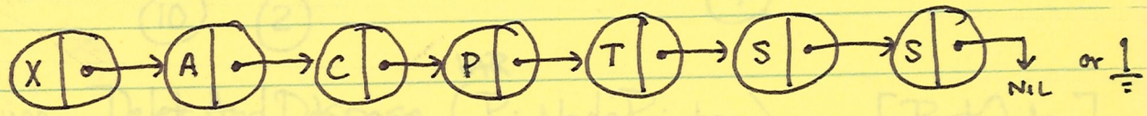
2. A. AB+C*DE/+F+

B. A*B+C+D/(E+F)

C.

2C.



3. LIST → 

AVAIL = 10.

(↓ NIL ⇔ ⏚)

4.
```
While not end of input do
    read (ch)
    case { ch in digit: push (S, ord(ch) - ord('0'))
           ch in oper: { b ← pop(S)
                         a ← pop(S)
                         case { ch = + : push (S, a+b)
                                ch = - : push (S, a-b)
                                ch = * : push (S, a*b) }
end while
```

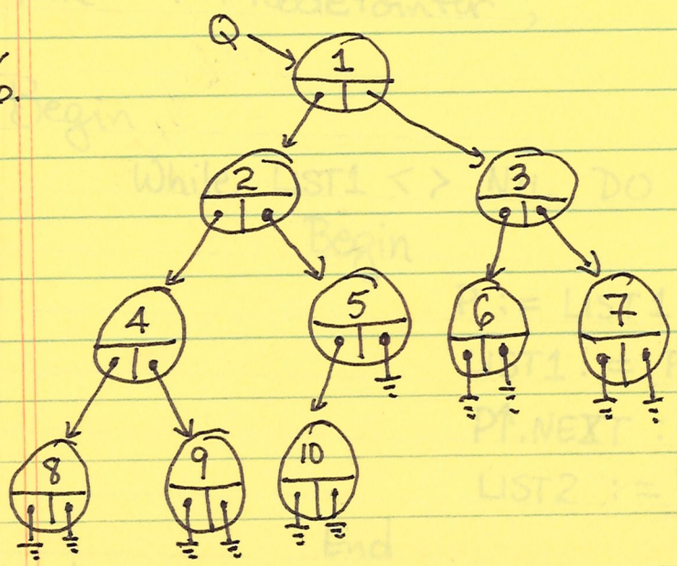
5

5A. next(p) ← next(q) ; next(q) ← p ; q ← p

B. r ← next(q) ; next(q) ← next(r)

C. next(p) ← p ; q ← p

6.



When i = 1 ABC is a recursive routine which builds a completely balanced tree with n nodes.

End;

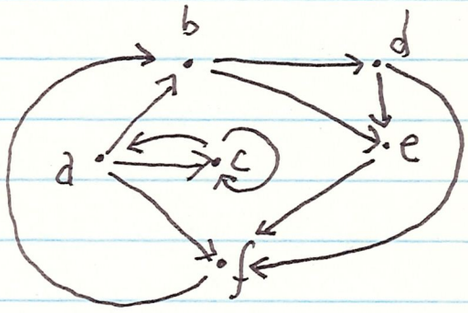
Answers

1A. (i) $O(n^2)$ (ii) $O(n \log n)$ (iii) $O(n^2)$ (iv) $O(n \log n)$ (v) $O(n \log n)$

B. $4n^2 \le 100n \iff 0 \le -4n^2 + 100n = 4n(25-n)$ hence SortA faster than SortB if $n < 25$ (or $n \le 25$)

2. A To right.

B. F ← 1 + trunc(6 * Random)

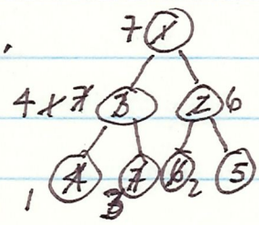


3 A. (1, 9, 1, 5, 8, 6, 4)

(1, 8, 1, 4, 9, 6, 5)

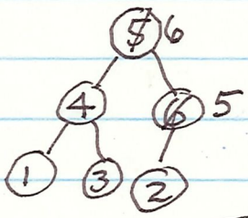
B. A 1 B 2 C 3 D 5 E 7 F 11 G 14 H 22 I 29

4 A.



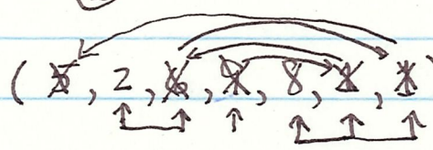
so (7, 4, 6, 1, 3, 2, 5)

B.



so (6, 4, 5, 1, 3, 2, 7)

5A



(1, 2, 1, 5, 8, 9 6)

B. 990 | 050 | 505 | 555 | 005 | 505 | 059 | 009 | 999 | 909

6. A $\sum_{1}^{n-1} k = \frac{(n-1)n}{2} = \binom{n}{2}$

B. C is a Tally of how many "wins" in comparing A_n element to the other n-1 elements. +1 (so): C[3] = 5 means X[3] is the 5th smallest element of X[1..n]

	In	Sevice	Out	Queue Time	System Time
7.&8:	2	4	6	0	4
	8	10	18	0	10
	10	4	22	8	12
	13	1	23	9	10
	16	3	26	7	10
	27	2	29	0	2
				24	48

A. $48/6 = 8$ min

B. $24/6 = 4$ min

C. idle 0-2; 6-8, 26-27, 29-30

$6/30 = 3/20 = 15/100$ .15 .20

D. $24/30 = 6/10 = .6$ .8

9

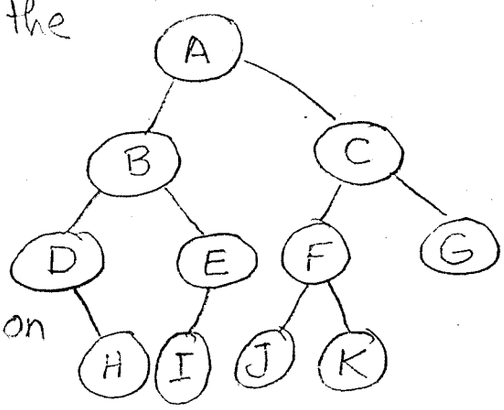
```
Program Thwart (input, output, data);

Var   i: integer;
      data: file of integer;

Begin
    Rewrite (Data);
    For i := 1 to 128 do
        Begin
            Data↑ ← i;
            put (Data)
        end
end.
```

(optional)

10. While not empty(S) or node ≠ nil do
    - if node ≠ nil then
      - Push(S, node)
      - node ← left(node)
    - ~~else~~
    - else
      - node ← Pop(S)
      - visit(node)
      - node ← right(node)

COP 3101 Pascal 2 Test 1 EACH Problem Worth 10 points

1. List the nodes of the tree to the right in the order given by



A. PreOrder B. PostOrder
C. InOrder

2. A Write the infix expression (A+B)*C+D/E+F in PostFix

B. Write the postfix expression AB*C+DEF+/+ in Infix

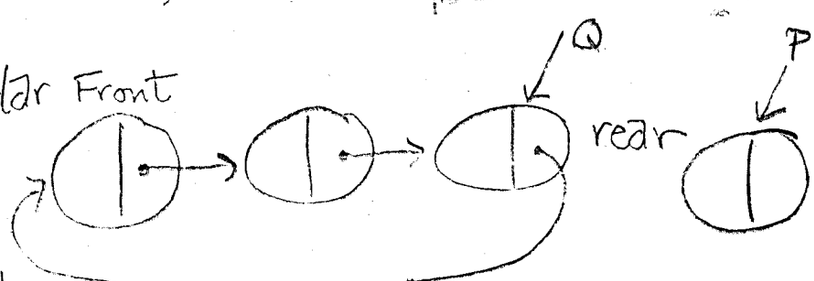
C. Build the BST for 3, 5, 4, 8, 7, 9, 1, 6, 2, 10.

3. For the "data box" array to the right. Draw the LIST or LISTS using node pictures (info | next) and give the value of AVAIL

1	A	5
2		4
3	T	7
4		0
5	C	12
6	S	0
7	S	6
8	X	1
9		11
10		9
11		2
12	P	3

4. Stack Evaluation of ~~Postfix~~ Postfix Expressions. Using the primitives ~~Push~~, ~~Pop~~ Push(S,A), POP(S) etc and using Algorithmic language write the algorithm for evaluating a postfix expression. Assume Input is either a single digit (i.e. all numbers are 1 digit long) or one of the chars +, - or *.

5. Q is a NodePointer pointing to a circular Front List which is being used as a queue.



A. P points to a node to be added to the rear of the rear of the queue. Assuming the queue is non-empty, list the pointer changes (in the correct order) needed to do this.

B. R is a nil pointer which is to receive the node at the front of the queue. Assuming the queue has at least two nodes, list the pointer changes (in the correct order) needed to do this

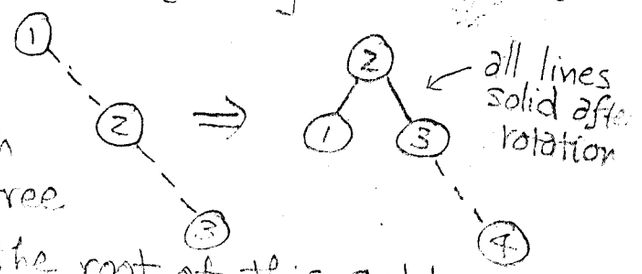
C. Repeat part A, this time assuming the Queue is empty.

R → NIL

6. Using nodes like  Construct the picture of what is created by the procedure call ABC(1, 10, Q).
What does this procedure do and what is the role of the parameters i and m?

```
Procedure ABC(i,m: integer; P node VAR
  if i≤m then { new(P)
                info(P) ← i
                ABC(2i, m, left(P))
                ABC(2i+1, m, right(P))
        else P ← nil
```

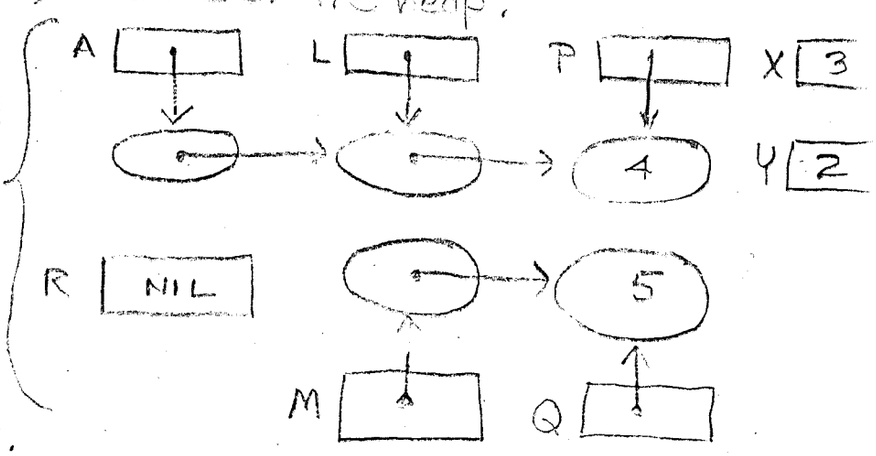
7. Construct the AVL tree for 6, 8, 5, 3, 9, 10, 2, 4
Your pictures should show the tree before each rotation you may add nodes to the rotated tree by using dashed links
example AVL for 1, 2, 3, 4



8. Write a recursive procedure (in algorithmic Language) which will delete and dispose the subtree of a [illegible] given a pointer to the root of this subtree.

9. Pointers: X, Y: integers; P, Q, R: ↑integers; L, M: ↑↑integer; A: ↑↑↑int
Rectangles are global, ovals are on the heap.
Our starting picture is to the right. Draw a similar picture after the sequence

A↑ ← M; R ← P;
P↑ ← Y; Y ← 7;
NEW(P); M ← L;
L ← A↑; P↑ ← M↑↑.



And List the current values of A↑↑↑, M↑↑, L↑↑, P↑, Q↑, R↑, X, Y

10. Using PASCAL write a procedure with 10 or fewer statements and no more than 3 local variables (each of which must be of type NodePointer) which will input LIST1 and LIST2 (=nil) and output LIST1 = NIL and LIST2 will be list1 in reverse order. Your Procedure must have correct syntax and make no procedure or function calls. (NODE = RECORD info: integer; next: NodePointer

if input is LIST1 → 1 → 2 → 3 → NIL LIST2 → NIL

output is LIST1 → NIL LIST2 → 3 → 2 → 1 → NIL

Pascal 2 test 2 Show All Work for Credit. Use Algorithmic (Excepting) Language
All Problems Worth 10 points.

1. A. Using "Big Oh" notation how many comparisons does it take to sort an n element array using
(i) Bubblesort? (ii) Quicksort on the Average?
(iii) Quicksort's worst case? (iv) Heapsort's worst case?
(v) Heapsort on the average?

B. SortA takes $4n^2$ time and SortB takes $100n$ time. For which values of n is SortA faster than SortB

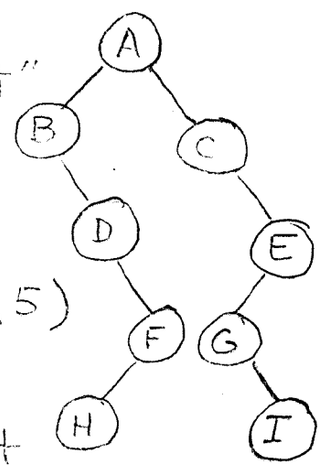
2. A. To the right is the adjacency list of a digraph. Draw the digraph

B. Using a function RANDOM which gives a random real in the interval (0,1) create a function F which returns a random integer i with $1 \le i \le 6$.

a: b, c, f
b: d, e
c: a, c
d: e, f
e: f
f: b

3. A. (SHELLSORT) If the initial array is (1, 9, 1, 5, 8, 6, 4) and the initial increment is 3 what is the resulting array after the first "pass" or "increment" of Shellsort

B. Give the "level-order" number of each node of the Binary Tree to the right.



4. (HEAPSORT) The initial array is (1,3,2,4,7,6,5)
A. Show the ARRAY (not a tree) after the creation of the initial heap
B. Show the ARRAY (not a tree) after the next heap (i.e. one "pass") is constructed

5. A (QUICKSORT) If the initial array is (5,2,6,9,8,1,1) show the resulting array after a call to REARRANGE (lb=1, ub=7) using the first entry as the test value

B. (RADIXSORT) The initial Queue is given below show the resulting Queue after one pass of radix sort

059	790	505	555	009	005	505	999	909	050

6. Consider the following "distribution sort" Initially X[1..n] is the array to be sorted, C[1..n] contains all ones, and Y[1..n] is undefined. Assume the values in X are distinct.

```
FOR i ← n downto 2 do FOR j ← i-1 downto 1 do
    if X[i] < X[j] then C[j] ← C[j]+1 else C[i] ← C[i]+1;
FOR i ← 1 to n do Y[C[i]] ← X[i]
```

(Hint $\sum_{i=1}^{m} i = \frac{m(m+1)}{2}$)

A. EXACTLY how many comparisons does this sort make?

B. What does C[1..n] contain at the end? That is if C[3]=5 what does mean about the relation of X[3] to the other values of X?

7. & 8. (SIMULATION) to the right is the arrival and service times for a 30 minute single channel single server system. Compute

A. Average System Time

B. Average Waiting Time (time in Queue)

C. Percent of the time the server is idle

D. The Average Queue Length.

Arrival Time	Service time
2	4
8	10
10	4
13	1
16	3
27	2

9. (FILES) [[ANSWER REQUIRES A COMPLETE PASCAL PROGRAM WITH CORRECT SYNTAX]]

Horrors! You have found out that your roommate is taking your girl(boy) friend out to impress ~~she~~ her (~~he~~ him) with the speed with which Quicksort can sort. Fortunately you have a plan to thwart this attempt at your romantic bliss.

Your roommate's Quicksort reads the EXTERNAL file DATA to create the array to be sorted. DATA was created as a PASCAL FILE OF INTEGER and contains 128 random integers.

Write a Pascal PROGRAM called THWART which will change Data to contain the integers 1 through 128 (without using the operation write) in increasing order.

10. (Removing Recursion) Rewrite the recursive procedure "inorder" below in an equivalent but non-recursive procedure "inorder2". Use a stack of nodes S and algorithmic Language and stack primitives Push, Pop etc. You may assume empty(S) is initially true.

```
Inorder (node)
    If node ≠ nil then { inorder ( left (node))
                         visit (node)
                         inorder ( right (node))
```

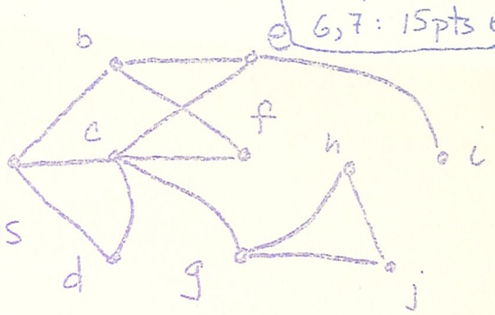
[Hint: Be a "run-time stack"]

(node is a usual for a binary tree)

PASCAL 2 FINAL PART 2 SHOW ALL WORK FOR CREDIT | 1-5 10 pts EACH | 6,7: 15 pts EACH

1 AB For the graph to the right find a spanning tree by A. DFS B. BFS (note there is exactly one correct answer)

C. For the tree below list the vertices in postorder

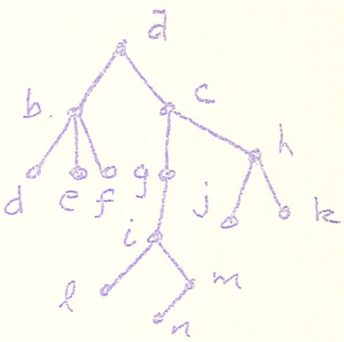


2. A. For the same tree to right draw the corresponding binary tree.

B.C. For the flow network below

B. "Label" as many nodes as possible

C. Show the edges and the changes that result when we use these labels to increase the flow.

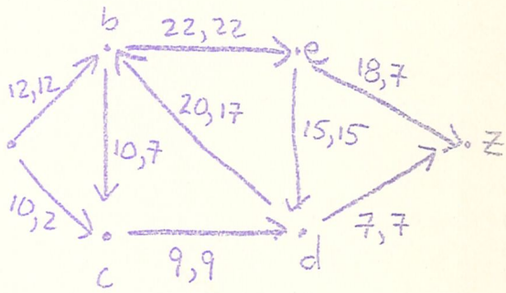


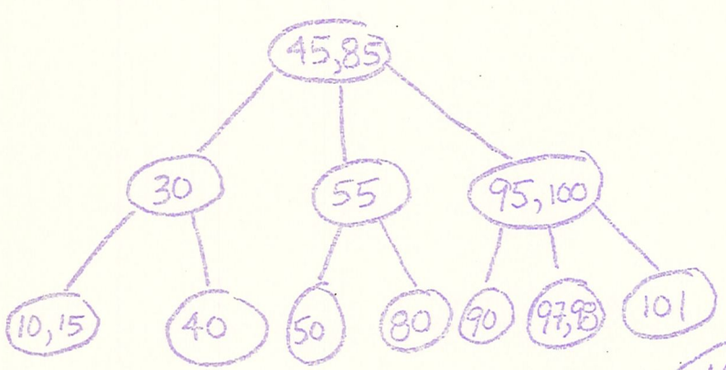
3. For the 2-3 tree below show the resulting 2-3 tree when

A. 75 is inserted

B. when 25 is inserted (to tree below not the one in A)

C. when 99 is inserted (to tree below not the one in A or B)

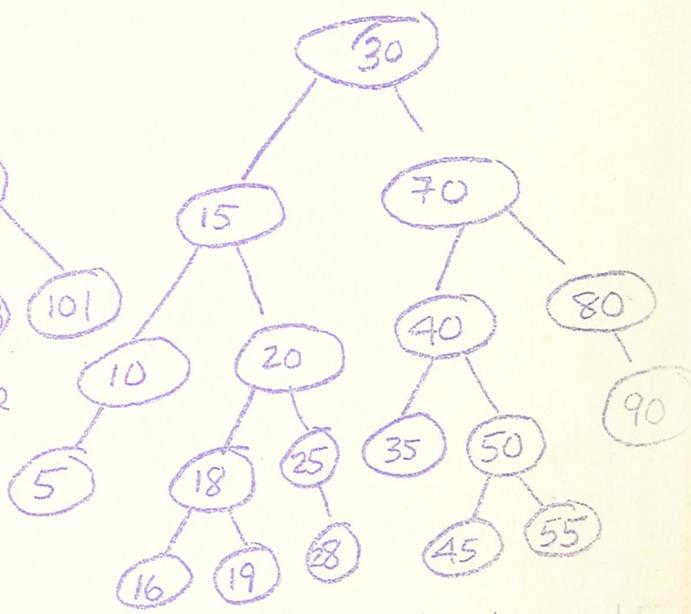




4. For the AVL tree to right show the resulting AUL tree when

A. 65 is inserted

B. when 17 is inserted (to tree to right not the one in A)



5. Hashing into an array TABLE [0..4] with hash(key) = key mod 5 and using linear probing to resolve hash collisions. Use 'E' to indicate an empty cell and 'D' for a deleted entry. The table is initially empty.

A. The following operations are done in sequence. For each tell how many positions in the hash table are checked before the operation is completed: (a) INSERT 39 (b) INSERT 60 (c) INSERT 79 (d) DELETE 60 (e) INSERT 96 (f) INSERT 99 (g) DELETE 39 (h) DELETE 79

B. Show the resulting table at the end of the sequence in A.

C. The average number of positions checked for the

6. Binary Search Define $b_k = n$ if each binary seach (sucessful or not) for any KEY in a sorted array $X[1..n]$ requires inspection of at most k entries in the array X but some binary search of a sort array with one more element (i.e. $X[1..n+1]$) requires inspection of at least $k+1$ elements of the array.

A. Explain why $b_1 = 1$ (in particular why it isn't larger or smaller)

B. Explain why $b_{k+1} = 1 + 2b_k$

C. Show the function $b_k = 2^k - 1$ satisfies the equation in A and the equation in B.

D. Obtain an Exact Answer, to the number of entries inspected in the worst case for binary search of the sorted array $X[1..n]$ in terms of n. (Hint the ceilling function $\lceil \ \rceil$ is helpful.) (~~not~~ $O(\log n)$ isn't exact enuff)

7. Below is "code" for a straight selection sort. It has two loops (labeled inner & outer) Find ASSERTIONS for

A The inner loop B. The outer loop

$X[1..n]$ the array to be sorted

```
i ← n
while i > 1 do                          OUTER LOOP
    j ← 1
    large ← X[1]
    index ← 1
    while j < i do                      INNER LOOP
        j ← j + 1
        if X[j] > large
            then large ← X[j]
                 index ← j
    X[index] ← X[i]
    X[i] ← large
    i ← i - 1
```