Midterm

(75 minutes open book exam)

Problem 1. (10 = 7 + 3 points). Consider sorting a linear array $A[1..n]$ with BUBBLESORT implemented with a while loop as follows.

```
x = TRUE;
while x = TRUE do
  x = FALSE;
  for i = 1 to n − 1 do
    if A[i] > A[i + 1] then
      A[i] ⟷ A[i + 1]; x = TRUE
    endif
  endfor
endwhile.
```

(a) Is it true that after j iterations of the `while`-loop the largest j items are in the correct last j positions of the array? Justify your answer.

(b) What does your answer to Question (a) imply for the running time of the algorithm?

SOLUTION TO PROBLEM 1. (a) Yes, it is true. Since the comparisons within a single iteration of the `for`-loop proceed from left to right, the largest item moves all the way to the last array position. Subsequent iterations leave that item at its correct position. During the second iteration move the second largest item moves to the second to the last array position, and so on. This implies that after i iterations the largest i items have moved to the last i array positions.

(b) After at most $n-1$ iterations of the `for`-loop the $n-1$ largest items are at their correct array positions, and thus all n items are. The n-th iteration makes no further swap. Bubble sort thus stops after at most n iterations and thus after at most $O(n^2)$ time.

Problem 2. (10 = 5+5 points). Recall that in a full binary tree every internal node has two children. All other nodes are leaves.

(a) How many different full binary trees with at most five leaves are there?

(b) Draw all different full binary trees with at most five leaves. [To save time and paper show only the internal nodes.]

SOLUTION TO PROBLEM 2. (a) We proved in class that the number of full binary trees with n leaves is the $(n-1)$-st Catalan number, which is $C_{n-1} = \frac{1}{n}\binom{2n-2}{n-1}$. We have $C_0 = 1$, $C_1 = 1$, $C_2 = 2$, $C_3 = 5$, $C_5 = 14$. The number of different trees is therefore $1 + 1 + 2 + 5 + 14 = 23$.

(b) The trees are shown in Figure 67.

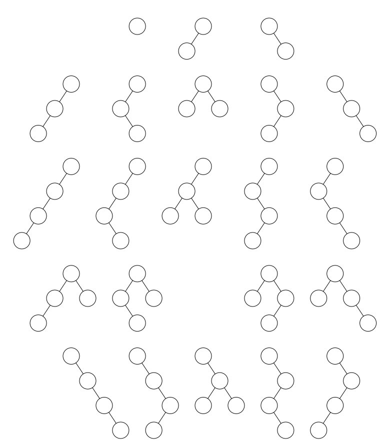

Figure 67: Only the internal nodes of the 23 trees are shown. Can you see the single leaf tree?

Problem 3. (10 points). Consider a binary search tree constructed by the successive insertion of $n-$

1 distinct keys (without rebalancing the tree). Is it true that if you pick a random sequence then each one of the possible $\frac{1}{n}\binom{2n-2}{n-1}$ binary trees is equally likely? Justify your answer.

SOLUTION TO PROBLEM 3. No, not every binary tree is equally likely. Note for example that there are six permutations of three keys but only five different binary trees. The balanced tree is twice as likely than each one of the four possible unbalanced trees.

Problem 4. ($10 = 5 + 5$ points). Construct the optimal Huffman tree and the corresponding optimal binary code for the following alphabet of ten letters with frequencies shown in parentheses: $A(5)$, $B(6)$, $C(9)$, $D(2)$, $E(11)$, $F(5)$, $G(6)$, $H(14)$, $I(4)$, $J(2)$. [Make sure that the weight of the root is the sum of frequencies, which is 64.]

(a) Show the Huffman tree.

(b) Show the binary code in a table.

SOLUTION TO PROBLEM 4. The Huffman tree is shown in Figure 68. To get the corresponding code shown in Table 4 we interpret each left edge as a 0 and each right edge as a 1.

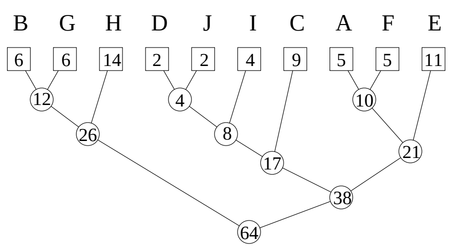

Figure 68: Huffman tree.

A	1100	F	1101
B	000	G	001
C	101	H	01
D	10000	I	1001
E	111	J	10001

Table 4: The optimal binary code.

Problem 5. ($10 = 5 + 5$ points). Let $p_0 = 5, p_1 = 6, p_2 = 7, p_3 = 1, p_4 = 10, p_5 = 2$, and for $1 \le i \le 5$ let X_i be a matrix with p_{i-1} rows and p_i columns. Let $X = X_1 \cdot X_2 \cdot X_3 \cdot X_4 \cdot X_5$.

(a) Determine the minimum number of elementary multiplications needed to compute X.

(b) Show the optimum parenthesization for computing X.

SOLUTION TO PROBLEM 5. (a) The minimum number of elementary multiplications is 102 and computed by dynamic programming with intermediate results shown in Figure 69.

p:

5	6	7	1	10	2

M:

	1	2	3	4	5
1	0	210	72	122	102
2		0	42	102	74
3			0	70	34
4				0	20
5					0

S:

	1	2	3	4	5
1		1	1	3	3
2			2	2	3
3				3	3
4					4
5					

Figure 69: Tables of partial results.

(b) The optimum parenthesization of the five matrices is $(X_1 \cdot (X_2 \cdot X_3)) \cdot (X_4 \cdot X_5)$, as illustrated in Figure 70.

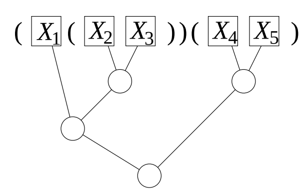

Figure 70: Optimal parenthesization.