

Data Structures
Spring 2005, Midterm1 Exam, March 29
7:00-9:00pm

1. (16%) Write both recursive and iterative C (or C-like) functions, *factorial*(*n*), to compute factorial of *n*.
2. (10%) Let arrays *X* and *Y* hold *m* and *n* sorted numbers, respectively. Write a C (or C-like) procedure *merge*(*X*, *Y*, *Z*, *m*, *n*), which merges the numbers in *X* and *Y* and produces a sorted list in *Z*. The worst case time complexity of the algorithm is $O(m + n)$.
3. (14%) Give the worst case time complexity of the following procedures as a function of *n* by using "big oh" notation.

(a) xxx (int : *n*)
{
 int *i, j, k, m, n, total*;
 i = 1 ;
 j = 0 ;
 total = 0 ;
 while (*i* <= *n*) {
 i = 4 * *i*;
 j = *j* + 2 ;
 }
 for (*m* = 1 ; *m* <= *i*; *m*++)
 for (*n* = 1 ; *n* <= *j* ; *n*++)
 total = *total* + *n* + *m*;
}
(b) yyy (int: *n*)
{
 int *i, j, x, y*;
 x = 0 ;
 y = 0 ;
 i = 1 ;
 while (*i* <= *n*) {
 if (*i* >= (*n* / 2))
 then for (*j* = *i*; *j* <= *n*; *j*++)
 x = *x* * 2 ;
 else for (*j* = 1; *j* <= *i*; *j* ++)
 y = *y* + 1 ;
 i = *i* + 1 ;
 }
}

4. (10%) Assume a row major representation of the array $A[1..6, 0..10, 1..20]$ with one word per element. Let 3000 be the address of $A[2, 4, 3]$. What is the address of $A[1, 3, 9]$?
5. (30%)
- (a) Stack can be utilized to evaluate a postfix expression and to translate an infix notation to postfix. Give two more applications that use stack as an auxiliary data structure.
 - (b) A linear list is being maintained circularly in an array $C[0..n-1]$ with *front* and *rear* set up as for circular queues. Give the condition to test if the queue is full.
 - (c) Place the following functions into asymptotically ascending order: n^2 , $\log n$, $\log n!$, 2^n .
 - (d) Design a data structure to represent a 3-variable polynomial $A(x, y, z)$.
 - (e) What is a FIFO ?
 - (f) Translate the following expression to its postfix notation: $a + b / c * d - e + f * g$
6. (10%) Translate the following infix notation to postfix using stack. Show the state of the stack after each update of the stack.

$$a - (b * c + d) * e$$

7. (10%) Write a recursive C (or C-like) procedure *max-min* to find the largest and the smallest numbers in a set of n different integers stored in an array A . The procedure, *max-min* ($A, i, j, fmax, fmin$), has 5 parameters. The first stores the numbers, the second and the third are the indices between which numbers are searched, and the fourth and the fifth return the maximum and minimum numbers. The procedure uses only $\frac{3}{2} \times n - 2$ comparisons.