Fall 2003

CMSC 420
Samet

ASSIGNMENT NUMBER 1

1. In the lecture we showed how two stacks can be made to share one segment of memory when using sequential allocation. Can you do the same for two queues, or a queue and a stack? If not, explain why.

2. Give all the possible ways of topologically sorting the set of relations A<B, F<B, D<B, G<F, G<E, E<D, F<D, E<C, D<C, B<C.

3. The algorithm we described in class for implementing the topological sort makes use of a queue to keep track of the nodes whose PRED_COUNT field has become zero and hence are ready to be output. Recall that these are the nodes whose direct successors have not yet been output. Can you use a stack instead of a queue? Give such an algorithm using pseudo-algol notation.

4. Suppose that ptr1 and ptr2 point to the last elements of disjoint circular lists l_1 and l_2, respectively. The lists l_1 and l_2 are interpreted as having their elements to the left of ptr1 and ptr2, respectively. In other words, NEXT(ptr1) and NEXT(ptr2) point to the first elements of l_1 and l_2, respectively. l_1 is inserted to the right of l_2 as follows:

```
if ptr2 ≠ λ then
  begin
        if ptr1 ≠ λ then NEXT(ptr1) ↔ NEXT(ptr2);
        ptr1 ← ptr2;
        ptr2 ← λ;
  end;;
```

   What happens if both ptr1 and ptr2 point to nodes in the same circular list?

5. How would you represent a circular list in such a way that it can be traversed efficiently in both directions, yet only use one link field per node. Hint: Note that if you are given two pointers to successive nodes a_{i-1} and a_i, then you can locate nodes a_{i-2} and a_{i+1}. You may also wish to manipulate the bit representation of the pointers.

6. Explain why it is difficult to implement a general deque with a singly-linked list.

7. Define a *tridiagonal matrix* M as a matrix whose diagonal elements are non-zero as are the diagonals to its left and right. All remaining elements are zero. Formally, element $M[i,j]$ is non-zero when $|i-j| \leq 1$ for $1 \leq i,\ j \leq n$. Show that there is a way to allocate memory sequentially for a tridiagonal matrix M so that only $3n-2$ locations are occupied. In particular, find constants c_0, c_1, c_2 for an access function of the form

$$location(M[i,j]) = c_0 + c_1 i + c_2 j \qquad |i-j| \leq 1.$$

8. Suppose that the elements of a two-dimensional array are ordered in such a way as to make it easy to expand the array's size by powers of 2 without having to move existing elements. In particular, the doubling occurs in cycles so that we double the size across the different dimensions in a cyclic manner (assume, without loss of generality, that we double the number of columns before we double the number of rows). For example, suppose that the array is initially of size 4×4. Thus the array is subsequently expanded to be 4×8, 8×8, 8×16, 16×16, etc. For the 4×4 array, we have the following order:

0	1	4	6
2	3	5	7
8	9	10	11
12	13	14	15

While for the 4×8 array, we have the following order:

0	1	4	6	16	20	24	28
2	3	5	7	17	21	25	29
8	9	10	11	18	22	26	30
12	13	14	15	19	23	27	31

What is the access function for element $A[i, j]$ assuming a $2^n \times 2^m$ square array where m and n differ by at most one and that the first element is $A[0, 0]$ and is located at location 0? The key to determining the location at which $A[i, j]$ is stored is to determine the expansion step m at which $A[i, j]$ was allocated (i.e., which doubling step). In particular, you must distinguish between even and odd doubling steps (i.e., $m = 2k$ or $m = 2k + 1$). For example, element $A[1, 2]$ is allocated in the third doubling step (i.e., $m = 3$) at which time elements 4–7 (i.e., $A[0, 2]$, $A[1, 2]$, $A[0, 3]$, $A[1, 3]$) were allocated, while element $A[2, 2]$ was allocated in the fourth doubling step (i.e., $m = 4$) at which time elements 8–15 (i.e., $A[2, 0]$, $A[2, 1]$, $A[2, 2]$, $A[2, 3]$, $A[3, 0]$, $A[3, 1]$, $A[3, 2]$, $A[3, 3]$) were allocated. Your answer should be in terms of i, j, and k.