2) (10 pts) DSN (Binary Heaps)

A **mininum heap** is typically implemented with an array, with the root node (**minimum value**) being stored in index 1 of the array. To insert a new value into a heap, it's originally placed in the first open slot, followed by running a "percolate up" operation. Write a function that inserts a value into a heap in this manner. You may assume that the array is allocated to be big enough to store the newly inserted value. The function prototype is as follows:

```
void insert(int* heap, int curSize, int newVal);
```

`heap` is a pointer to an array which currently stores `curSize` number of values (but has room for at least 1 more). `newVal` is the new number to be inserted into the heap. Write this function which inserts the value newVal into this **minimum heap**. Take care to avoid infinite loops or array out of bounds issues. You may assume that index curSize+1 is in bounds for the array heap. Also, remember that index 0 of the array heap is unused. **You may not write any helper functions.**

```
void insert(int* heap, int curSize, int newVal) {

    heap[curSize+1] = newVal;
    int idx = curSize+1;
    while (idx > 1) {
        if (heap[idx/2] < heap[idx]) break;
        int tmp = heap[idx/2];
        heap[idx/2] = heap[idx];
        heap[idx]   = tmp;
        idx /= 2;
    }
}
```

Grading: 1 pt place newVal in array slot curSize+1
1 pt – some loop or recursion
1 pt – access array at current index divided by 2 (parent)
2 pts – triggering a swap if the two appropriate items are out of order
3 pts – swap mechanics
2 pts – correctly ending the loop