1) (5 pts) ANL (Algorithm Analysis)

What is the best and worst case runtime for the following algorithm, in terms of the input parameter n? You may assume that the array pointed to by arr is of length n. Give a brief explanation for your answers.

```
int foo(int * arr, int n, int value){

    int cur = 0, jump = n/2;
    while (jump > 0) {
        if (value > arr[cur])
            cur += jump;
        else if (value == arr[cur])
            return cur;

        jump = jump/2;
    }

    return cur;
}
```

The best case run time is O(1). It's possible that on the very first loop iteration that the else if clause that returns cur triggers. In this situation, only a fixed number of statements, all of which are simple, run.

The worst case run time is O(lg n). The number of times the loop runs is controlled by jump. Each time, jump's value divides by 2 and the loop will end the iteration after jump equals 1. Since jump starts out as n/2, if we let k equal the number of loop iterations, then we get the equation $(n/2) / 2^k = 1$. Solving for k in this equation yields $k = \log_2 n - 1$. Since the work in each loop iteration is constant, the run time of O(lg n) follows.

Grading: 1 pt for the best case answer, 1 pt for its justification, 1 pt for the worst case answer, 2 pts for its justification