1) (10 pts) ANL (Algorithm Analysis) What is the Big-Oh memory usage for the function call `createNode(N)`? Please provide your answer in terms of the input parameter, N. **Please justify your answer by either evaluating an appropriate recurrence relation or summation.**

```
typedef struct Node Node;
struct Node {
   Node ** children;
   int val;
};
Node * createNode(int N) {
   Node * res = (Node *) malloc(sizeof(Node));
   if (N == 0) return res;
   res->children = (Node **) malloc(sizeof(Node*) * N);
   res->children[0] = createNode(N / 2);
   res->val = 0;
   for (int i = 0; i < N; i++)
      res->val += i;

   return res;
}
```

The amount of memory produced by the function ignoring any recursive call is O(N). Taking into account the recursive call we find that the memory created fits the following recurrence relation

T(N) = T(N/2) + O(N)
T(1) = O(1)

Solving the recurrence relation we get the following,

T(N) = T(N/2) + N
T(N/2) = T(N/4) + N/2
T(N) = T(N/4) + N/2 + N
T(N/4) = T(N/8) + N/4
T(N) = T(N/8) + N/4 + N/2 + N

after *k* iterations

$$T(N) = T\left(\frac{N}{2^k}\right) + \sum_{i=0}^{k-1} \frac{N}{2^i} \leq T(1) + N\left(\frac{1}{1-\frac{1}{2}}\right) = 1 + 2N = O(N)$$

Alternatively, a student could recognize that at each level half as much memory will be allocated, so that ultimately, the amount of memory allocated will be roughly N + N/2 + N/4 + …, and solve the sum.

Grading: 8 pts to get to the correct sum (any way), 2 pts to evaluate it and give the correct answer. If the sum is incorrect, then max credit is 7 pts. Give partial based on work and breakdown of # of nodes created. If recurrence relation method is followed, 4 pts for stating the recurrence, 2 pts for iteration, 2 pts for its solution.