CSC 212: Data Structures and Abstractions
University of Rhode Island
Problem Set 02 (Fall 2016)

Stephanie Donnelly

1. (a)

$$O(nlogn)$$

(b)

$$O(n)$$

(c)

$$O(n2^n)$$

(d)

$$O(n2^n)$$

(e)

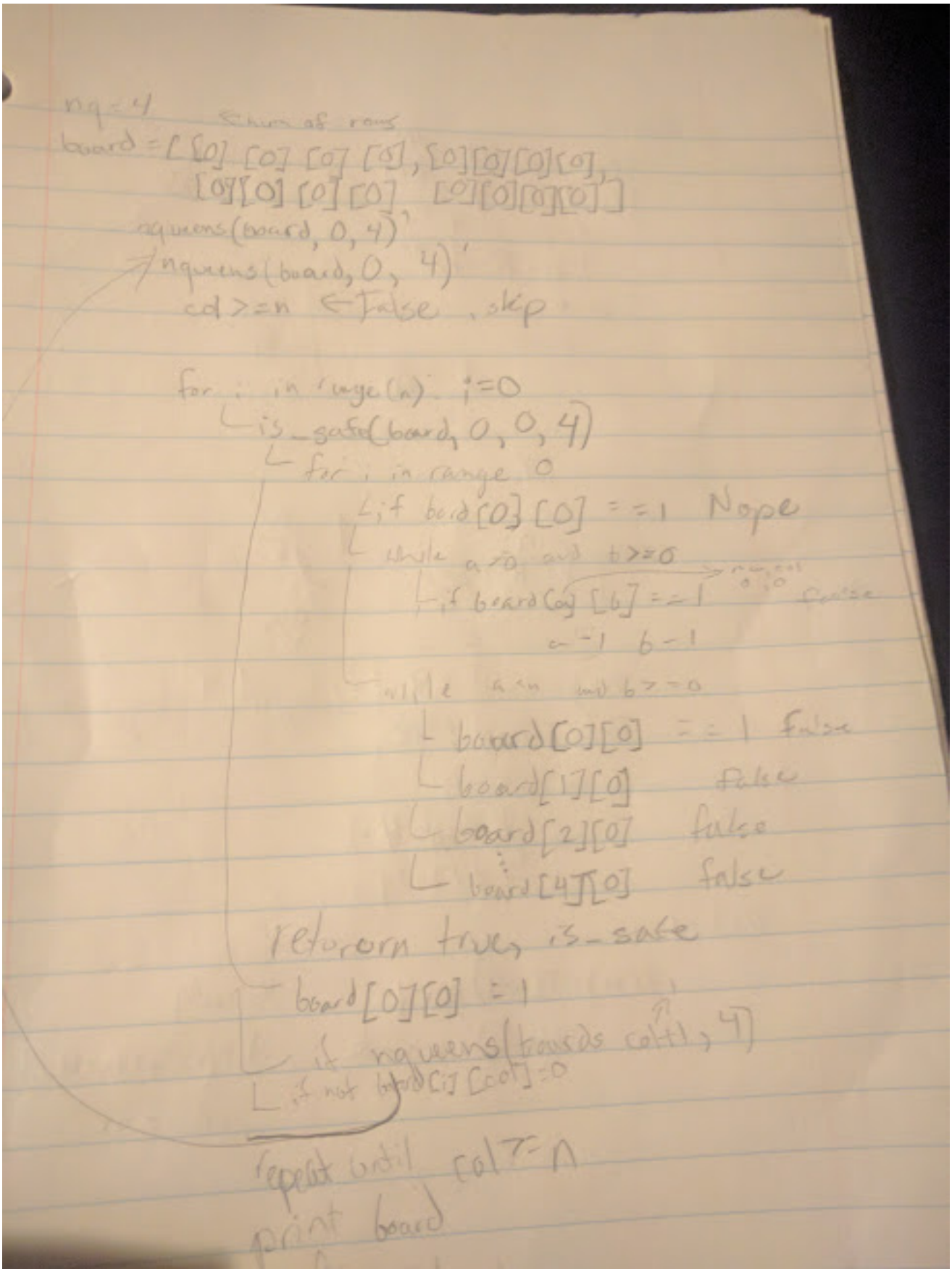
$$O(logn)$$

(f)

$$O(n^2)$$

2.

$$O(logn)$$

3. 
nq = 4
board = [[0] [0] [0] [0], [0] [0] [0] [0],
[0] [0] [0] [0] [0] [0] [0] [0]]
nqueens(board, 0, 4)
nqueens(board, 0, 4)
col >= n ← False, skip
for i in range(n): i=0
is_safe(board, 0, 0, 4)
for i in range 0
if board[0] [0] == 1 Nope
while a>0 and b>=0
if board[a] [b] == 1 false
a-1 b-1
while a<n and b>=0
board[0][0] == 1 false
board[1][0] false
board[2][0] false
board[4][0] false
return true, is_safe
board[0][0] = 1
if nqueens(board, col+1, 4)
if not board[i] [col] = 0
repeat until col >= n
print board

4. `listings` environment:

```
long int tribonacci(int n) {
    long int current = 0;
    long int prev= 0;
    long int prev_prev = 0;
    for(int i = 0; i<n; i++){
        current = num;
        num = current + prev + prev_prev;
        prev  = num;
        prev_prev = num - prev;
    }
}
```

5. `listings` environment:

```
void reverse(char *str, int n) {
    for(int i = 0; i<n/2; i++){
        char temp = *str;
        *str = *(str-i)
        *(str-i) = temp;
    }
}
```

6. `listings` environment:

```
        void reverse(char *str, int n) {
    for(int i = 0; i<n/2; i++){
        char temp = *str;
        *str = *(str-i)
        *(str-i) = temp;
    }
}

int palindrome(char *str, int n) {
    string s = "␣";
    while(*str != '\0'){
        s +=*str;
        str++;
    }
    int mid = strlen(s)/2;
    if(strlen(s)%2 ==1){
        mid++;
    }
    string half2 = reverse(s.substr(mid) //get from midpoint on and reverse
       that
    string half1 = s.substr(0, mid);
    if(half1 == half2){
        return 1;
    }else{
        return 0;
    }
}
```

7. `listings` environment:

```
void sort(int *array, int n) {
    for(int i = 0; i < n; i++){
        if(array[i]>array[i+1]{
                                int temp = array[i];
                                array[i+1] = array[i];
```

```
                array[i] = temp;
            }
        }
    }
```

8. (a) correct
   (b) compiler error, can't convert int to int*
   (c) compiler error: don't convert between double* and int*
   (d) correct
   (e) correct
   (f) correct
   (g) compiler error, can't convert int** to int*
   (h) correct
   (i) logical error: n will end up being the address of p2 (a pointer, which is probably useless)
   (j) compiler error, can't convert between double* and int*

9. 0 2 4 6 8 10 12 14 16 18