Bradley Lamitie

FINAL EXAM

1. Using iteration method, find a tight bound for the solution of the following recurrence equation.
   $T(n) = T(n/2) + 1$, $n > 1$. Assume $T(1) = 1$ (5 points).

$T(n) = T(n/2) + 1 \qquad T(1) = 1$

$T(n/2) = T(n/4) + 1$

$T(n/4) = T(n/8) + 1$

$T(n) = T(n/2) + 1$

$T(n) = (T(n/4) + 1) + 1$

$T(n) = (((T(n/8) + 1) + 1) + 1)$

$\ldots$ k times

$T(n) = T(n/2^k) + k$

Assume $n = 2^k$

$T(n) = 1 + k$

$\log_2 n = \log_2 2^k$

$\log n = k$

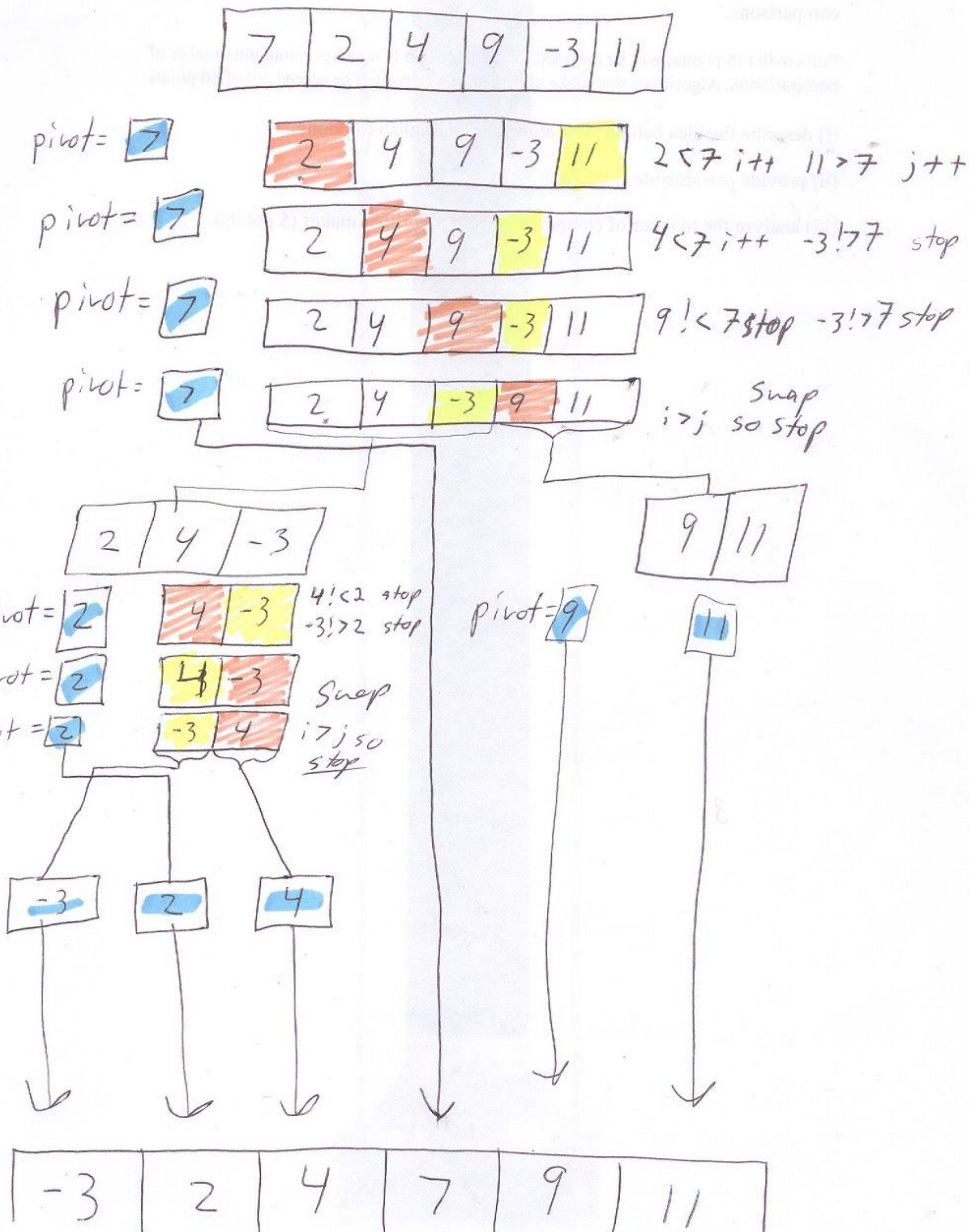
$T(n) = T(n/2^k) + k$

$T(n) = T(n/2^{\log n}) + \log n$

$T(n) = T(1) + \log n$

$T(n) = 1 + \log n \qquad O(\log n)$

2. Show step by step how the quicksort algorithm sorts the array 7, 2, 4, 9, -3, 11. Indicate at each step what the pivot is (5 points).



(i) We will do this by running through a tournament style comparison based off of min max. First we will find the maximum & second maximum of an array, and do the same to the other half. Then we compare the four results.

(ii)

```
Maximums(i, j, max1, max2){
    if i=j
      max1 = max2 = A[i]
    else{
      if (i=j-1){
        if (A[i] > A[j]){
          max1 = A[i]
          max2 = A[j]
        } else{
          max1 = A[j]
          max2 = A[i]
        }
    }else{
        mid = ⌊(i+j)/2⌋
        maximums(i, mid, max1, max2)
        maximums(mid+1, j, max3, max4)
    }
  if (max1 > max2){
    if (max1 > max3){
      if (max1 > max4){
        max = max1
      }else{
        max = max4
    }else{
      if (max3 > max4){
        max = max3
      }else{
        max = max4
      }
  }else{
    if (max2 > max3){
      if (max2 > max4){
        max = max2
      }else{
        max = max4
    }else{
      if (max3 > max4){
        max = max3
      }else{
        max = max4
      }
```

// 3

Known to take $\frac{3}{2}n - 2$ comparisons

maximum of 3 comparisons

(iii)

$\frac{3}{2}n - 2 + 3 = \boxed{\frac{3}{2}n + 1}$