Name: Georges Hatem

CS590 Homework 9: Divide-and-Conquer Application Exercises

Due Date: April 3, 2022

Problem 11.6.12:

Algorithm for maximum and minimum using divide-and-conquer

```
MaxMin(i, j, max, min)
{
if (i=j) then max = min = a[i] ;
else if (i=j-1) then {
if (a[i] < a[j]) then max = a[j] ; min = a[i];
else max = a[i] ; min = a[j] ;
}
else
{
mid = ( i + j )/2 ;
MaxMin( i, mid, max, min );
```

MaxMin(mid+1, j, max1, min1);

if (max < max1) then max = max1 ;

if (min > min1) then min = min1;

}

}

Maximum number of comparisons:

If T(n) represents this number, then the resulting recurrence relation is

n = 0 , n=1

T(n) = 1 n=2

= T(n/2) + T(n/2) + 2 n>2

When n is a power of two, $n = 2^k$ for some positive integer k, then

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

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

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

.

.

.

$= 2^{k-1} T(2) + \sum(1 \leq i \leq k-1) 2^k$

$= 2^{k-1} + 2^k - 2$

$= 3n/2 - 2 = O(n)$

Here the comparisons are not more than 3n/2