

Midterm 2

Daren Liu

1) double pl (double arr[lo..hi]) {

double max_diff = 0;

for (int i = lo; i < hi; i++) {

for (int j = i + 1; j < hi; j++) {

if (arr[j] - arr[i] > max_diff && arr[j] != arr[i])
max_diff = (arr[j] - arr[i]);

}

}

return max_diff;

}

$$h - (lo + 1) + 1$$

$$(n) \sum_{i=lo}^{hi} \sum_{j=i+1}^{hi} (1) = \sum_{i=lo}^{hi} (hi - (i + 1) + 1) = \sum_{i=lo}^{hi} (hi - i)$$

$$\sum_{i=lo}^{hi} hi - \sum_{i=lo}^{hi} i = hi(hi - lo + 1) - \frac{hi(hi + 1)}{2}$$

$$hi = n$$

$$(n) \in \Theta(n^2)$$

44
50

2) double p2 (double arr[lo..hi]) {

mergesort(arr, lo, hi);

return arr[hi] - arr[lo];

}

Mergesort run-time is $\Theta(n \log n)$

3) double p3 (double arr[lo..hi]) {

if (arr[lo] == 1 && arr[hi] < 0)

return arr[lo];

if (arr[hi] < 0)

return arr[hi] + p3(arr[lo..hi-1]);

else

return p3(arr[lo..hi-1]);

}

$C(1) = 1$

$C(n) = 1 + C(n-1) \quad n \geq 1$

$C(2) = 1 + C(1) = 2$

$C(3) = 1 + C(2) = 3$

$C(4) = 1 + C(3) = 4$

Guess $C(n) = n$

$1 + C(n-1) = 1 + n - 1 = n = C(n)$

$C(n) \in \Theta(n)$

4) double p4 (double arr[lo..hi]) {

7

if (arr.size() == 1 && arr[lo] < 0)

return arr[lo];

else if (arr[lo] < 0) return 0

int m = arr.size() / 2;

double lsum = p4(arr[lo..m]);

double rsum = p4(arr[m+1..hi]);

return lsum + rsum;

3

$$C(1) = 1$$

$$C(n) = 1 + 2 \left(C\left(\frac{n}{2}\right) \right)$$

$$C(n) \in \mathcal{O}(n \log n)$$

n

5) bool p5(int a, int b) {

1

if (b == 0) {

if (a == 1)

return true;

return false

3

return p5(b, a % b);

3

analysis ?