Gregory Byrne Test Review HW

HW: show each pass of the insertion sort

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 9 | 6 | 8 | 6 | 2 | 4 |
| 1 | 3 | 9 | 6 | 8 | 6 | 2 | 4 |
| 1 | 3 | 6 | 9 | 8 | 6 | 2 | 4 |
| 1 | 3 | 6 | 8 | 9 | 6 | 2 | 4 |
| 1 | 3 | 6 | 6 | 8 | 9 | 2 | 4 |
| 1 | 2 | 3 | 6 | 6 | 8 | 9 | 4 |
| 1 | 2 | 3 | 4 | 6 | 6 | 8 | 9 |
| 1 | 2 | 3 | 4 | 6 | 6 | 8 | 9 |
|  |  |  |  |  |  |  |  |

HW: Do the first pass of quicksort. Show the partition and show which two calls to quicksort would happen next

pivot = (x[L] + x[R]) / 2

pivots = 8+1 / 2 = 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 7 | 5 | 6 | 3 | 4 | 2 | 1 |
| 1 | 7 | 5 | 6 | 3 | 4 | 2 | 8 |
| 1 | 2 | 5 | 6 | 3 | 4 | 7 | 8 |
| 1 | 2 | 4 | 6 | 3 | 5 | 7 | 8 |
| 1 | 2 | 4 | 3 | 6 | 5 | 7 | 8 |

fill in the recursive calls….

quicksort(x, 0, 3)   
quicksort(x, 4, 7)

HW: Show makeheap, the first part of heapsort, which turns this array into a maxheap. Use the mapping we discussed in class, and start with n/2 going to down to 0 for the order of the nodes.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 9 | 1 | 6 | 1 | 7 | 3 | 8 | 5 |

HW: Binary search

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 6 | 7 | 7 | 8 | 9 |

show binarySearch(x, 0, 7, 5) (search list x from location 0 to 7 for target = 5)

L = 0 mid = 3 R = 7

L = 0 mid = 1 R = 2

L = 2 mid = 2 R = 2

Target: 5 Not Found in Data

show binarySearch(x, 0, 7, 7)

L = 0 mid = 3 R = 7

L = 4 mid = 5 R = 7

Found the target: 7 at position: 5

Optional HW: Write pseudocode for binarySearch

int binary\_search(int x[], int left, int right, int target)

{

if (left > right) {

return -1;

}

int mid = (left + right) / 2;

if (x[mid] == target) {

return mid;

}

if (x[mid] > target) {

return binary\_search(x, left, mid - 1, target);

}

return binary\_search(x, mid + 1, right, target);

}

Golden Mean Search

f(x) = 9 - x2

find x of the max to the nearest .1

goldenMean(f, -2, +3)

L = -2, R = +3

phi = 1.618 = (sqrt(5) + 1) / 2

S = (R - L) / phi = 3.09

a = R - S = 3 - 3.09 = -.09

b = L + S = -2 + 3.09 = + 1.09

R = 1.09

b = -.09

S = (R - L) / phi = 1.909

a = R - S = 1.09 - 1.909 = -.81

L = a = -.81

HW: Golden Mean

f(x) = 14 - (x+1)2 = 13 - x2 - 2x

find x of the max to the nearest .1

goldenMean(f, -3, +5)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PHI | L | R | S | a | b | ya | yb | abs (b-a) |
| 1.618034 | -3 | 5 | 4.944272 | 0.055728 | 1.944272 | 12.88544 | 5.331263 | 1.888544 |
| 1.618034 | -3 | 1.944272 | 3.055728 | -1.11146 | 0.055728 | 13.98758 | 12.88544 | 1.167184 |
| 1.618034 | -3 | 0.055728 | 1.888544 | -1.83282 | -1.11146 | 13.30642 | 13.98758 | 0.72136 |
| 1.618034 | -1.83282 | 0.055728 | 1.167184 | -1.11146 | -0.66563 | 13.98758 | 13.8882 | 0.445825 |
| 1.618034 | -1.83282 | -0.66563 | 0.72136 | -1.38699 | -1.11146 | 13.85024 | 13.98758 | 0.275535 |
| 1.618034 | -1.38699 | -0.66563 | 0.445825 | -1.11146 | -0.94117 | 13.98758 | 13.99654 | 0.17029 |
| 1.618034 | -1.11146 | -0.66563 | 0.275535 | -0.94117 | -0.83592 | 13.99654 | 13.97308 | 0.105245 |
| 1.618034 | -1.11146 | -0.83592 | 0.17029 | -1.00621 | -0.94117 | 13.99996 | 13.99654 | 0.065045 |

max = (-1, 14)