**Question 1**(2 points)

*Saved*

Match each recurrence relation on the right to a solution on the left.

Question 1 options:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1234 | O(n2){"version":"1.1","math":"\(O(n^2)\)"} | | 1234 | O(n){"version":"1.1","math":"\(O(n)\)"} | | 1234 | O(nlog23){"version":"1.1","math":"\(O(n^{log\_2 3})\)"} | | 1234 | O(nlogn){"version":"1.1","math":"\(O(n \log n)\)"} | |  | |  |  | | --- | --- | | **1**. | T(n) = 4T(n/2) + O(n) | | **2**. | T(n) = 3T(n/2) + O(n) | | **3**. | T(n) = 2T(n/2) + O(n) | | **4**. | T(n) = T(n/2) + O(n) | |

**Question 2**(2 points)

Which of the following insights is NOT used in the development of the divide-and-conquer algorithm dicussed in the week 2 lecture?

Question 2 options:

|  |  |
| --- | --- |
|  | When considering pairs of points lying on opposite sides of line l, we only need to consider those points whose distance form line l is less than delta. |
|  | The closest pair of points are not necessarily on the same (left or right) side of line l (slides 67-72) |
|  | Before the recursive divide-and-conquer procedure is even started, we must construct two different arrays containing the input points: one array holds the points in order by x-coordinate and the other in order by y-coordinate. |
|  | Every set of n points has the property that when sorted by y-coordinate, points that are close to each other in the ordering are also close to each other on the plane. |
|  | Each one of the 16 equal sized dashed squares shown on slides 74-75 may contain one point at most. |

**Question 3**(2 points)

*Saved*

Not including the memory space necessary to hold the input array A, how much extra memory space is used by MergeSort and by QuickSort? Choose all that apply.

Question 3 options:

|  |  |
| --- | --- |
|  | Θ(1){"version":"1.1","math":"\(\Theta(1)\)"} extra memory space is used by QuickSort |
|  | Θ(1){"version":"1.1","math":"\(\Theta(1)\)"} extra memory space is used by MergeSort |
|  | Θ(n){"version":"1.1","math":"\(\Theta(n)\)"} extra memory space is used by QuickSort |
|  | Θ(n){"version":"1.1","math":"\(\Theta(n)\)"} extra memory space is used by MergeSort |

**Question 4**(2 points)

*Saved*

Which of the following algorithms covered in lecture 2 have a worst case running time of O(logn){"version":"1.1","math":"\(O(\log n)\)"}? Choose all that apply.

Question 4 options:

|  |  |
| --- | --- |
|  | Binary Search |
|  | MergeSort |
|  | PingalaPower |
|  | QuickSort |
|  | FastMultiply |

**Question 5**(2 points)

*Saved*

The initial call to multiply x=1,234,567 and y=9,876,543 using function FastMultiply on page 41 of your textbook will make three recursive calls to FastMultiply. What three pairs of values will the three recursive calls be multiplying?

Question 5 options:

|  |  |
| --- | --- |
|  | 1234 and 9876, 567 and 543, 667 and 9333 |
|  | 12 and 98, 34567 and 76543, -34555 and -76445 |
|  | 123 and 543, 4567 and 9876, 4444 and 9333 |
|  | 1234 and 6543, 567 and 987, 667 and 5556 |
|  | 123 and 987, 4567 and 6543, -4444 and -5556 |