1. If the AiBj+1 and Ai+1Bj is considered as right tree and left tree, the algorithm of finding max points can be think as binary tree traversal. Firstly, we visit all right positions in current row. If there is no available position in the right, we start moving down. When the final position is arrived, we compare the points we get in path with max point, if we collect more point than max point, change max point. After that, we go one step back and try other possible position and apply same procedure until all possible paths are visited. It has O((m*n) ^2) time complexity because in each iteration there are 2 ways to go and m times right and n times down movement are necessary. There is no early exit in this algorithm, and all paths must be visited to find path contains max point, so best, worst and average case time complexities are the same.

2. To find the median of an array with decrease and conquer algorithm, solution of the kth smallest element in array is used. Because the median is the middle element or elements of a sorted list of numbers. So, to find the middle element of an odd size array, we must find (n/2)th element of the array. On the other, if the array size is even number, we must find both (n/2)th element and the next element.

   The working principle of solution of kth smallest element in array is that last element of the array which is called pivot is picked and the elements which is less than pivot placed to left side of the pivot, the elements which are greater than pivot placed to right side of the pilot. This procedure is provided by partition algorithm, and it returns the position of the pivot. After that, if the position of the pivot is equal to k, returns the pivot element. If the position of the pivot is greater than k, kth smallest element procedure is applied to left side of the pivot. Otherwise, it is applied to right side of the pivot.

The time complexity of finding median of array is O(n^2). Because partition algorithm has O(n) time complexity and kth smallest element algorithm runs n times in the worst case. Worst case occurred in this case that in every iteration of kthSmallestElement algorithm, size of the array decreases by only 1.

3.

a. The working principle of findWinner algorithm is that, continue traversing the circular linked list until one element is left. In every iteration remove first from the left and continue with second from the left. The time complexity of this algorithm is O(n). Because the loop runs n-1 times, and it removes an element in each iteration. Removing next elements in circular linked list takes constant time.

b. In findWinner algorithm, players are stored in array. The working principle of findWinner algorithm is that, firstly, number of players is found by length of the array, and it is converted from decimal into binary. After that, leftmost digit of the binary is placed on the right of the binary number. Then, the winner is decimal version of the current binary number.

The worst-time time complexity of findWinner algorithm is O(logn). Because converting a decimal to binary is takes logn time. In every iteration, the number is divided into 2.

4. Time complexity of ternary search is better than binary search. Because array is divided into 3 parts instead of 2 parts in ternary search. In this way, the size of the part of the array which contains key, is n/3. Finding an element in n/2 size array is harder than in n/3. But there is one thing which cannot be seen in time complexities. In big O notation, constants are not taken in account. For example, in each iteration of ternary search, there are twice comparisons as much as binary search. For this reason, if the array is divided into n parts instead of 3 parts, key must be compared with each element in the array, and it becomes linear search which has O(n) time complexity.
5.
    a. If the key is in middle position – not middle of the array, it has special formula to find middle of the array by considering values which the array contains – it is found in first iteration. So, the time complexity of this case is O (1).
    b. In binary search, no matter what are the values which array contains, the key element is compared with the middle element of the array. But in interpolation search, values also are considered. For example, let array is [1, 2, 3, 15, 16, 17, 18, 19, 21] and key is 3. In binary search, 2 is compared with 16 first. On the other hand, 2 is compared with 1.