

هر سوال را در محل در نظر گرفته شده پاسخ دهید. پاسخ های خارج از محل تصحیح نمیشوند. شماره دانشجویی باید با اعداد لاتین نوشته شود.

۱. [۴۰] کدام یک از گزینه های زیر *RecursiveCall* مناسبی برای *QuickSort* می باشد. توضیح دهید.

```
public static void quickSort(int[] arr, int low, int high)
{
    int pivot;
    if(high>low)
    {
        pivot = partition(arr, low, high);
        quickSort(arr, low, pivot-1);
        quickSort(arr, pivot+1, high);
    }
}
```

(آ)

```
public static void quickSort(int[] arr, int low, int high)
{
    int pivot;
    if(high<low)
    {
        pivot = partition(arr, low, high);
        quickSort(arr, low, pivot-1);
        quickSort(arr, pivot+1, high);
    }
}
```

(ب)

```
public static void quickSort(int[] arr, int low, int high)
{
    int pivot;
    if(high>low)
    {
        pivot = partition(arr, low, high);
        quickSort(arr, low, pivot);
        quickSort(arr, pivot, high);
    }
}
```

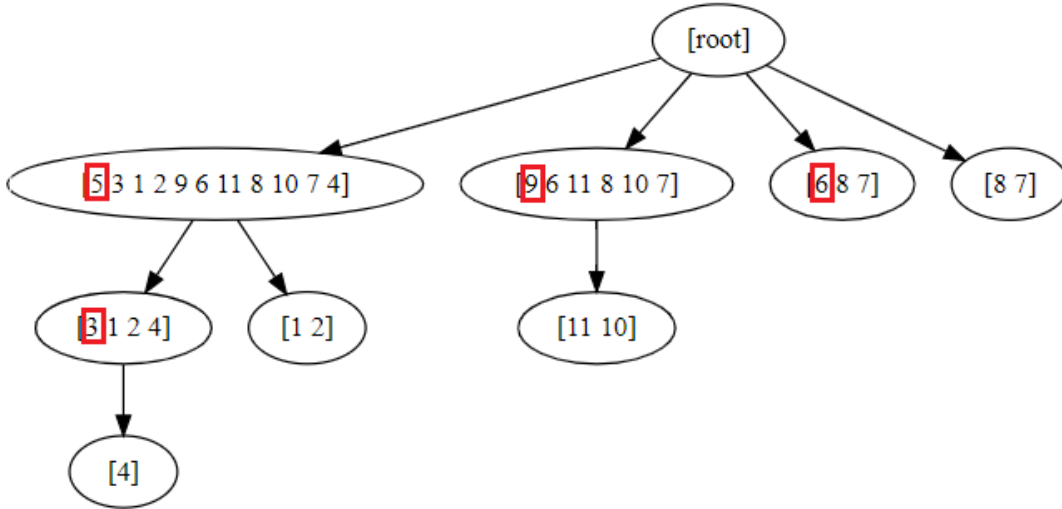
(ج)

(د) هیچ کدام

طبق مطالب گفته شده و اسلاید ها گزینه (آ) صحیح می باشد. گزینه (ب) به دلیل شرط *if* و گزینه (ج) هم به دلیل تقسیم بندی اشتباه آرایه (به تابع *partition* را به عنوان ورودی می دهد نه  $partition + 1$  و  $partition - 1$ ) غلط هستند. content...

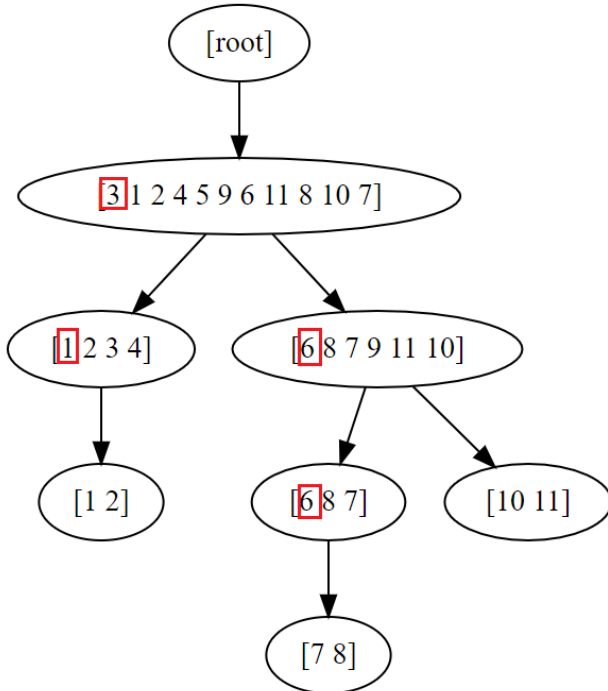
۲. [۶۰] درخت بازگشتی الگوریتم QuickSort را برای آرایه زیر با استفاده از تکنیک Tail Recursion متوازن و انتخاب اولین عنصر به عنوان pivot رسم کنید.

$$A = [5, 3, 1, 2, 9, 6, 11, 8, 10, 7, 4]$$



شکل ۱: Using Tail Recursion technique

فقط جهت اطلاع:



شکل ۲: Normal quicksort: Without Tail Recursion. Just FYI