Solutions for Microsoft Questions On-Campus (2023)

Ans 1 :

```
public boolean dfs(ArrayList<ArrayList<Integer>>
graph,              int curr, boolean vis[], boolean
stack[]) {
   vis[curr] = true;
   stack[curr] = true;

   for(int i=0; i<graph.get(curr).size(); i++) {
      int neigh = graph.get(curr).get(i);
      if(stack[neigh] == true) {
         return true;
      }
      if(!vis[neigh] && dfs(graph, neigh, vis,
stack)){
         return true;
      }
   }

   stack[curr] = false;
   return false;
}

public boolean canFinish(int numCourses, int[][]
prerequisites) {
   int n = numCourses;
```

```
    ArrayList<ArrayList<Integer>> graph = new
ArrayList<>();
    for(int i=0; i<numCourses; i++) {
        graph.add(new ArrayList<>());
    }

    for(int i=0; i<prerequisites.length; i++) {
        int v = prerequisites[i][0];
        int u = prerequisites[i][1];
        graph.get(u).add(v);
    }

    boolean vis[] = new boolean[n];
    boolean stack[] = new boolean[n];

    for(int i=0; i<n; i++) {
        boolean isCycle = dfs(graph, i, vis, stack);
        if(isCycle) {
            return false;
        }
    }

    return true;
}
```

Ans 2 :

```java
public static int solution(String S) { //O(N)
   int n = S.length();

   if(n < 2) {
      return 0;
   }
   int left[] = new int[n-1];
   int right[] = new int[n-1];

   left[0] = S.charAt(0) != '>' ? 1 : 0;
   for(int i=1; i<n-1; i++) {
      if(S.charAt(i) != '>') {
        left[i] = left[i-1]+1;
      } else {
        left[i] = 0;
      }
   }

   right[n-2] = S.charAt(n-1) != '<' ? 1 : 0;
   for(int i=n-3; i>=0; i--) {
      if(S.charAt(i+1) != '<') {
         right[i] = right[i+1]+1;
      } else {
        right[i] = 0;
      }
   }
```

```
    int ans = 0;
    for(int i=0; i<S.length()-1; i++) {
      int len = 2*Math.min(left[i], right[i]);
      if(len > ans) {
       ans = len;
      }
    }

    return ans;
}
```