说明：1、请直接在答题纸上作答，题目中涉及的模板代码为 C#语言，如果使用其他编程语言，在答题纸注明编程语言并作答即可。
2、部分题目只需写结果和思路，请仔细读题。
3、本卷的【附加题】如果没有接触过 Unity3d 游戏引擎的同学可以不作答。

1. 快速排序的时间复杂度是多少？为什么？

2. 在某卡牌游戏中，玩家手中有一副点数卡牌，用整数数组 nums 表示。在游戏中，当给玩家提供一个目标点数 target 时，玩家需要找出卡牌中点数之和恰好为目标点数的两张卡牌。假设每组卡牌及对应的目标点数有且仅有一种答案。
请编写算法，帮助玩家找到对应的这两张卡牌，并以下标数组的形式返回。
【示例】
卡牌数组 nums = [3, 7, 10, 12]
目标点数 target = 10
【结果】
因为 nums[0] + nums[1] = 3 + 7 = 10，故返回[0, 1]

```
public int[] FindTwoCards(int[] nums, int target){

}
```

3. 现有一个游戏的地图被划分成 m*n 个格子，一些格子之间被墙隔开，该游戏的角色可以在没有墙的相邻两个格子之间移动，现在玩家想要知道角色是不是能够走到地图的某个位置。请补全下面的代码来实现这个功能。
注意，这个功能在游戏中需要多次查询，需要考虑效率问题。

题目模板使用的是 C#语言，如果使用其他编程语言，请在后面的 其他语言 处答题。

```
public class Solver{
    private int _m,_n;

    //初始化函数
    //m、n 表示地图的宽高
    public void Init(int m, int n){
        _m = m;
        _n = n;

    }

    //用以添加一堵墙的函数
    //(x1,y1)、(x2,y2)代表墙两边的两个相邻的格子
    public void AddWall(int x1, int y1, int x2, int y2){

    }


    //用以查询角色能否到达想要到达的位置的函数
    //roleX,roleY 表示角色当前位置，targetX,targetY 表示玩家想要到达的位置
    //返回是否可达
    public bool Solve(int roleX, int roleY, int targetX, int targetY){

    }
```

}

4. 湖边有五个人在钓鱼，每个人至多钓起了一条鱼，通过以下条件判断每个人钓起了什么鱼、位置顺序和穿着：C 穿着红色外套坐在最右边且钓到了鱼，A 坐在穿绿色外套的人旁边，D 钓到了鲷鱼，与 D 相邻的人没有穿着黑色和紫色外套，B 旁边的人钓到了鲤鱼，钓到鲢鱼的人坐在 E 的旁边，没有钓到鱼的人旁边的人穿着紫色外套，穿着黑色外套的人钓到了鲈鱼，穿着蓝色外套的人坐在最左边， E 钓到了鱼，B 和 E 旁边坐着两个人。（**只需写出最终答案即可**）

5. 现有一款模拟经营类游戏，玩家经营着一个食品生产工房。工房会收到很多来自 NPC 客人的食品生产订单。已知：
   1）每种食品生产订单都会有客人指定的截止交付时间，超过了截止交付时间则报酬会减少;
   2）每种食品生产订单都需要花去 cost 小时，而且不能中断；
   3）玩家当前能力等级同时只能进行一种食品加工。
   请你作为一名玩家，安排工房的加工生产的顺序，以减少总的订单超时时间。
   【输入描述】
   第一行包含一个正整数 n(1<=n<=20)，表示玩家收到的加工订单的数量。接下来 n 行，每行包含两个整数，分别表示第 i 种订单的 截止交付时间 和 加工时间 cost。
   【输出描述】
   累计总共需要超出截止时间多少小时才能完成所有的订单。

   输入例子:
   3
   3 3
   8 1
   3 2
   输出例子:
   2

   【例子说明】
   输入样例表示有 3 份订单。
   第一个订单截止时间在 3 小时后，需要 3 小时时间完成。
   第二个订单截止时间在 8 小时后，需要 1 小时时间完成。
   第三个订单截止时间在 3 小时后，需要 2 小时时间完成。
   因此合理的顺序是 1->3->2 或 3->1->2，均需要推迟 2 小时。
   此题只需要清楚的写出解题的关键思路即可。

6. 觉得自己之后在团队中的定位是什么?

【附加题】Unity3D 部分

1. Unity3d 环境下如何实现对象池（写出大体思路即可）？你认为一个塔防类游戏中哪些东西适合使用对象池？
2. 现有一个函数 ComplexCalculate, 调用时会返回一个随机的 Vector3 值，请使用协程实现补齐 Solve 函数，以实现以下功能：
   1）先等待 1s 后再进行计算；
   2）循环调用 30 次 ComplexCalculate 并求其结果的平均值。这 30 次计算分成 3 帧，每帧调用 10 次；
   3）最后把运算结果赋予 TargetPos 属性。

```
public class Solution : MonoBehaviour{
    // 目标位置
    public Vector3 TargetPos;
    // 带复杂位置计算的函数，可以直接调用
    private Vector3 ComplexCalculate(){
        Vector3 result;
        // ...
        // 内部复杂计算已实现
        // ...
        return result;
    }

    public void Solve(){

    }
    protected IEnumerator CalculateTargetPos(){

    }
}
```

3. 请将下面这段关于 Unity 新功能的描述的英文翻译成中文。
   Unity Technologies has shipped Unity 2021.1, the first of two scheduled updates to the game engine this year. Here, we’ve picked out some highlights for CG artists.
   1. Point Light Shadows in the Universal Render PipelineThis functionality enables the URP to generate more realistic shadows from local light sources, like torches or campfires.
   2. New Camera Sorting Layer Texture for 2D graphicsChanges to Unity’s 2D toolset include a new Camera Sorting Layer Texture, which combines pixel data into a frontmost sorting layer, and which is intended for creating effects like heat haze and refractive water.
   翻译：