Grupo Weboo
Universidad de La Habana

Segundo Examen de Programación
Curso 2014-2015

Instalando WiFi en la UH

NOTA: Si usted está leyendo este documento sin haber extraído el compactado que descargó del sitio, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios **no se guarden**. Si usted comete este error y entrega una solución vacía, no tendrá oportunidad de reclamar.

Se desea proveer al campus de la Universidad de la Habana de conexión inalámbrica a la red universitaria.

Para ello se ha hecho un estudio previo de la infraestructura existente y se han determinado los puntos específicos donde es posible instalar Access Points (AP) dentro del área del campus.

Se dispone de un conjunto de APs que tienen diferentes alcances y se desea obtener con ellos la máxima cobertura del área universitaria. El número de APs disponibles es mayor o igual que el número de puntos habilitados para su instalación.

Usted debe implementar un algoritmo que devuelva el área máxima que es posible cubrir instalando el AP conveniente en cada punto.

El área del campus se considerará rectangular y cada AP tiene asociado un alcance o radio de acción que por simplicidad será considerado como un cuadrado alrededor de la ubicación del punto. La Figura 1 nos muestra la ubicación de un AP de alcance 1 y la Figura 2 nos muestra un AP de alcance 2.

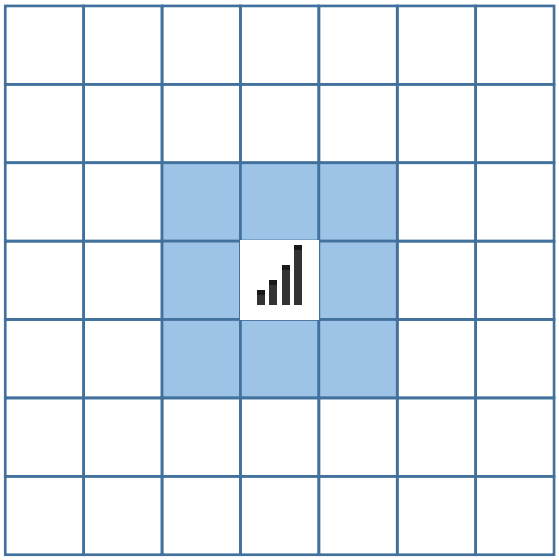

Figura 1 AP con alcance de radio 1

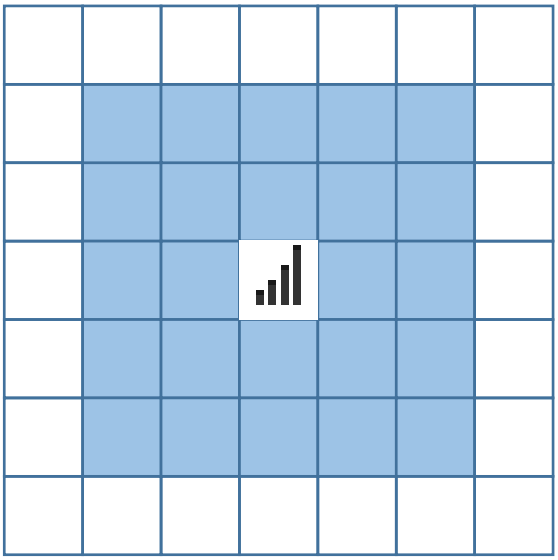

Figura 2 AP con alcance de radio 2

Note que según la ubicación, el alcance de un AP puede salirse de los límites del rectángulo (área a cubrir). La Figura 3 nos muestra la ubicación de un AP de radio 3 que sale del área. También note que el área cubierta por los APs puede solaparse. La Figura 4 nos muestra dos APs cuyos alcances (2 y 1) se solapan (lo que no quiere decir que la ubicación de ambos AP sea incorrecta siempre que valgan para cumplir el objetivo general.

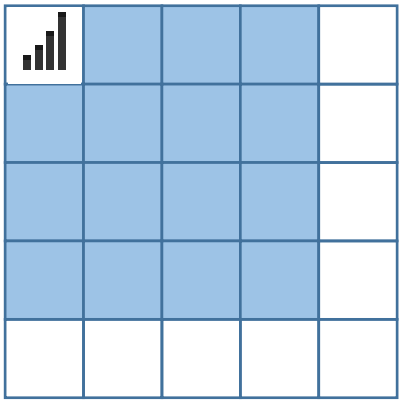

Figura 3 AP con alcance de radio 3, y que excede los límites del área.

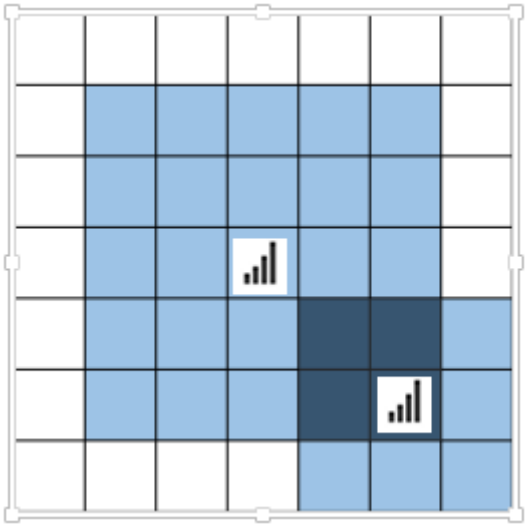

Figura 4 Dos APs, uno con alcance de 2 unidades y otro con alcance de 1 unidad cuyas señales se solapan (área más oscura).

El área a cubrir será representada por una matriz (array rectangular) de `bool` en la que los puntos donde pueden ser instalados los APs tendrán el valor `true`.

Note que:

- No es necesario colocar todos los APs disponibles para alcanzar la mayor cobertura.
- Pueden haber distintas distribuciones de APs colocados con las que se obtenga la misma mayor cobertura, en la respuesta solo interesa el dato de cuál es la mayor cobertura.

Ejemplo 1

En este ejemplo se tienen tres APs con rangos 2, 1 y 1 y tres puntos para su instalación. La Figura 5 representa una distribución donde se maximiza el área a cubrir. La solución es 40 (15 + 25). La Figura 6 representa una distribución donde no se maximiza el área (37).

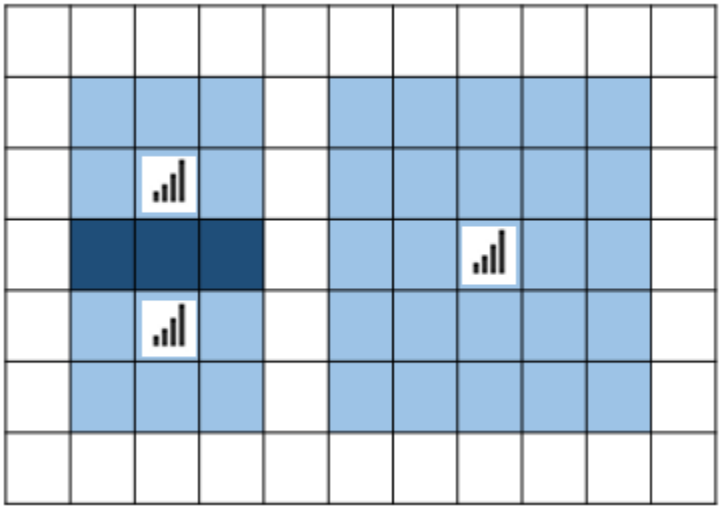

Figura 5

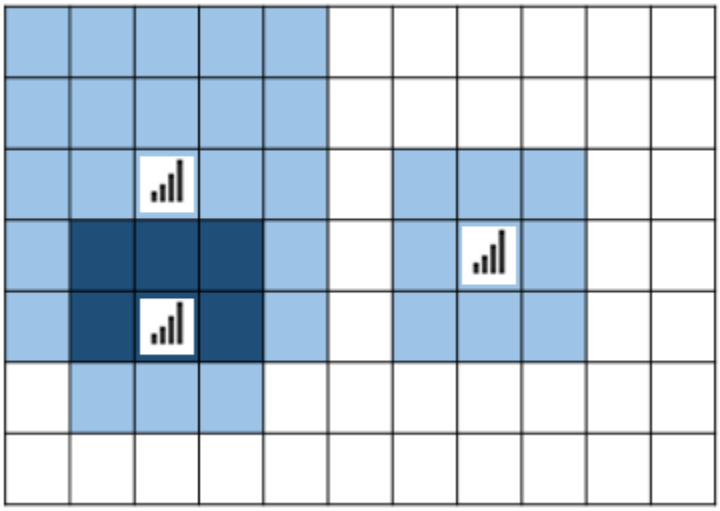

Figura 6

Ejemplo 2

En este ejemplo se tienen 3 APs con rango 2, 1 y 1 y solo dos puntos para su instalación. Las Figuras 6 y 7 representan distribuciones donde en ambos casos toda el área queda cubierta y la solución es 12. En la Figura 6 se ubican los dos APs de rango 1. En la Figura 7 se ubica un AP de rango 2 y otro de rango 1

(notar que en este caso, el AP de rango 1 queda completamente solapado, su instalación no contribuye al área cubierta, pero no se pide minimizar la cantidad de APs utilizados).

Figura 6

Figura 7

El problema

Implemente el método `CubrirArea` que debe devolver el valor del área máxima que se puede cubrir. El método recibe como parámetros un **array** de `int` con los alcances de los APs disponibles (este array nunca tendrá valores <= 0) y un **array** de `bool` que representa el área y que tendrá `true` en las casillas donde es posible instalar un AP. Si en **array** del área no existiera ningún valor true, el método debe devolver 0.

```
public static int CubrirArea(bool[,] area, int[] alcances)
        {
            // Borre esta línea e implemente su código aquí
            throw new NotImplementedException();
        }
```