Prácticas
Caso de Estudio 3

Solado económico (I)

- **Necesitamos solar una superficie cuadrada de n metros de lado. A tal efecto, contratamos a los hermanos Scott, unos instaladores reformistas con amplia experiencia, para realizar el trabajo. Para el solado podemos elegir diferentes baldosas cuadradas de s0, s1, s2, … metros de lado.**

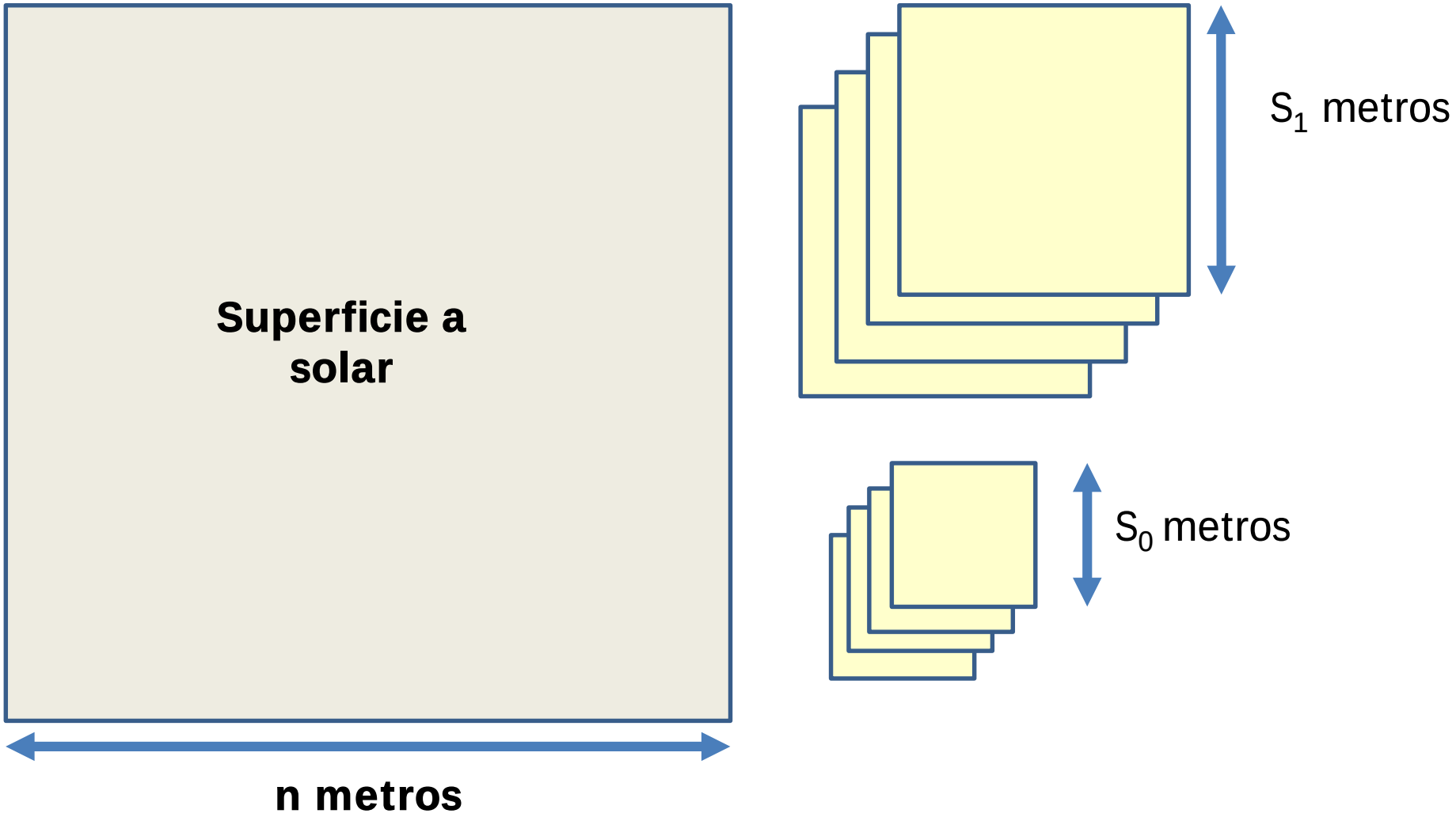
Gracias a sus contactos profesionales, los hermanos Scott pueden conseguir las baldosas al mismo precio por baldosa independientemente del tamaño. Además, pueden conseguir la cantidad necesaria de baldosas de cualquier tamaño. Para reducir costes, pretendemos utilizar tan pocas baldosas como sea posible y por razones estéticas queremos usar baldosas enteras (aunque se mezclen tamaños).

Solado económico (II)

- **Se pide, usando sólo baldosas enteras**
  1. **Desarrollar e implementar en java un algoritmo voraz que resuelva el problema anterior.**
  2. **Determinar la complejidad del algoritmo**
  3. **Indicar si el algoritmo es óptimo. Para ello, demuéstrelo o indique un contraejemplo.**

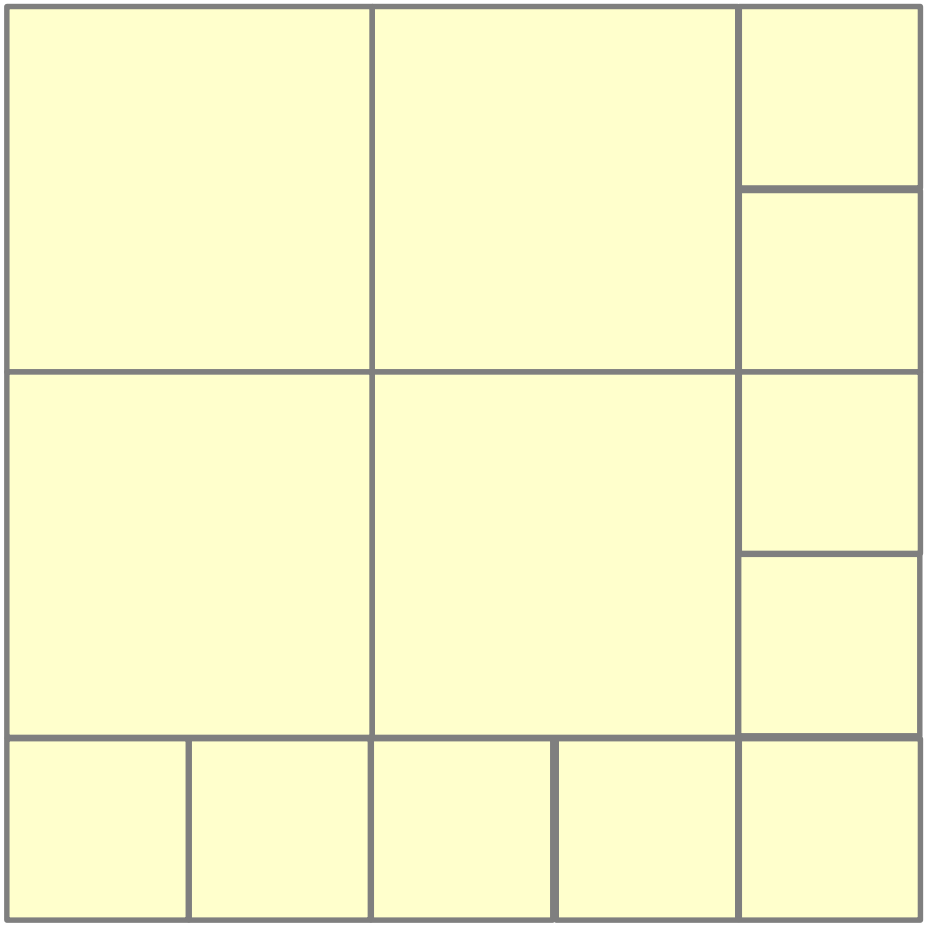
Solado económico (III)

- **El problema**



Solado económico (IV)

- **El problema (resultado)**



Solado económico (V)

- **Estrategia voraz**
  - **Ordenar las baldosas en orden decreciente**
  - **Solar la superficie de izquierda a derecha y de arriba hacia abajo usando tantas badosas como sea possible del tamaño mayor**
  - **Seguir con baldosas del siguiente tamaño y repetir el proceso hasta que no se puedan poner más baldosas**

- **¿Optimalidad?**

Solado económico (V)

- **Estrategia voraz**
  - **Ordenar las baldosas en orden decreciente**
  - **Solar la superficie de izquierda a derecha y de arriba hacia abajo usando tantas badosas como sea possible del tamaño mayor**
  - **Seguir con baldosas del siguiente tamaño y repetir el proceso hasta que no se puedan poner más baldosas**

- **¿Optimalidad?**
  - **Ver qué ocurre con un suelo de lado 4 metros y baldosas de tamaños 3, 2 y 1 metros**
    - › Solución del algoritmo: ????
    - › Solución óptima: ????

Solado económico (VI)

- **Algoritmo.**

```
Algorithm tileFloor (tiles, n)
    current ← 0
    set empty solution
    sort tiles in decreasing order
    while there is room for whole tiles do
        if room for size tile(current) then
            place tile of size tile(current)
            reduce available floor surface
            add one tile of size tile(current) to solution
        else
            current ← current +1
        end_if
    end_while
    return solution
```

Building up the solution

Selection and feasibility

Add current element to the solution

Solado económico (VII)

- **Sugerencias**
  - **Trabajar con unidades de 1 unidad lineal (cm, m,..)**
  - **Usar una variable para el lado del suelo: side**
  - **Usar un array para los tamaños de baldosa (tamaños enteros en orden creciente): tilesSizes[ ]**
  - **Usar un array de tamaño side x side para el suelo (cada elemento representa una celda de 1 x 1 unidades^2 en el suelo: floor[ ][ ]**

- **Ejemplo:**
  - **side = 3**
  - **tilesSizes = {1, 2, 3, 4}**
  - **floor[3][3] ← el lado es igual a 3**

Solado económico (VII)´

- Procesando el array floor

```
for (int i = 0; i < side; i++){
  for (int j = 0; j < side; j++){
    if (floor[i][j] == 0) {
-- The place is empty, do whatever you need to do --
                . . .
```

- Comprobando si hay sitio para una baldosa

```
current = tilesSize.length - 1;
size = tilesSize[current];
while(current >=0 &&
      !(i+size<=side && j+size<=side)){
   current--; size = tilesSize[current];
 }
```

Solado económico (VIII)

- **¿Complejidad?**
  - **¿Ordenación?**
  - **¿Bucles anidados?**