

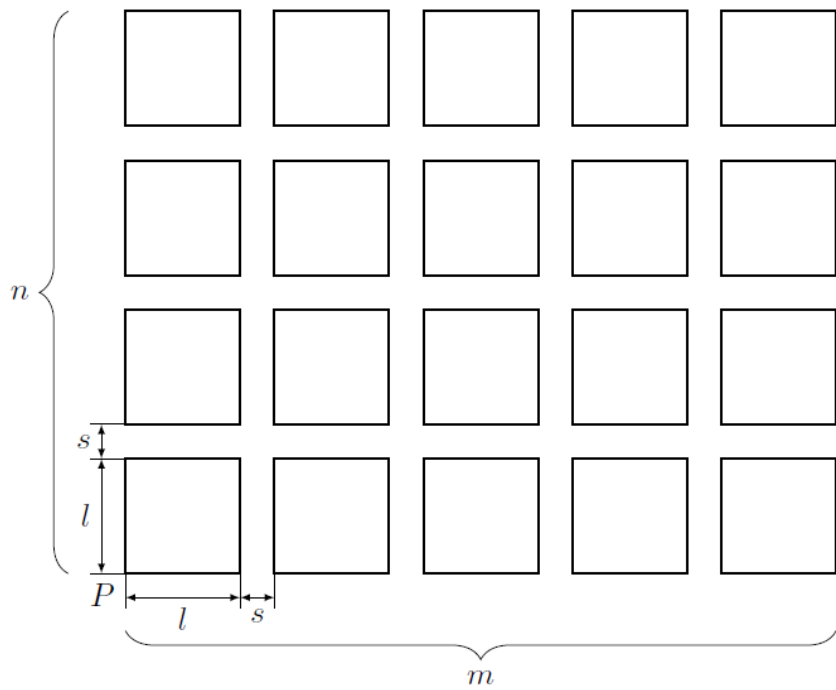


PRÁTICA 4

ALEATORIEDADE



Considere uma cidade organizada em termos de quarteirões. Para simplificar, vamos assumir que cada quarteirão será de forma quadrada e irá conter um único edifício. Cada edifício terá uma largura determinada pela largura do quarteirão e uma altura máxima imposta. Os edifícios serão separados uns dos outros por ruas com uma largura fixa.



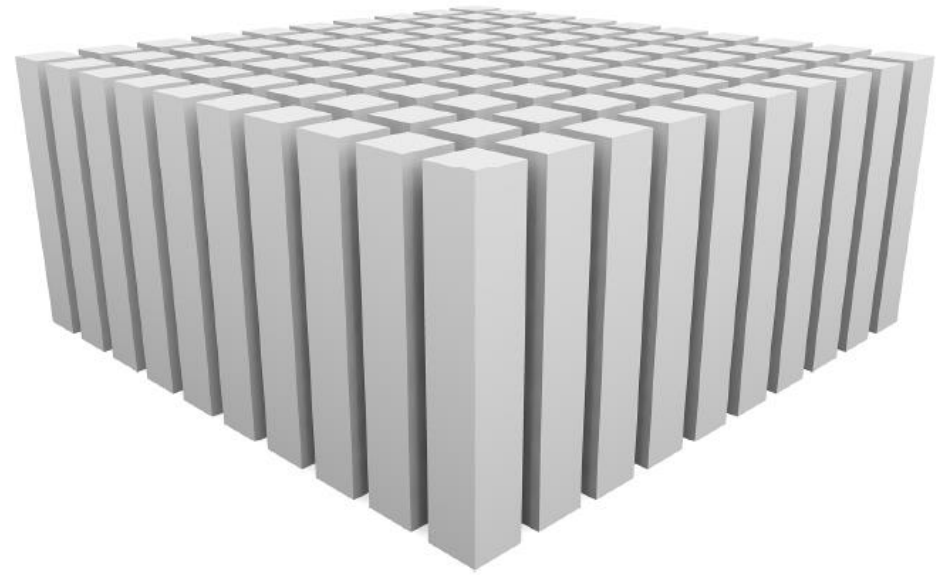
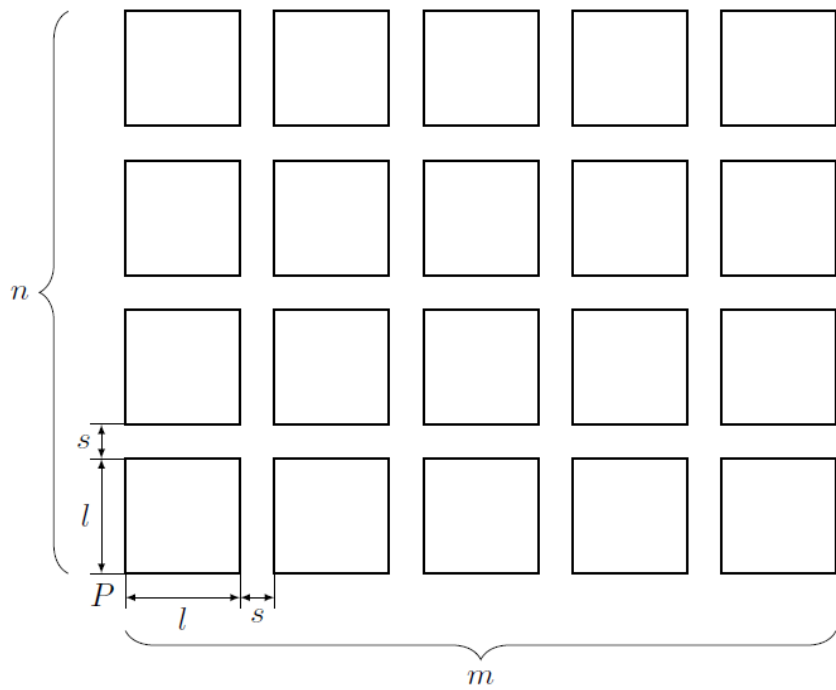
As duas seguintes funções modelam uma destas cidades:

```
malha_predios(p, n_ruas, m_predios, l, h, s) =
  if n_ruas == 0
    nothing
  else
    rua_predios(p, m_predios, l, h, s)
    malha_predios(p + vy(l + s), n_ruas - 1, m_predios, l, h, s)
  end
```

```
rua_predios(p, m_predios, l, h, s) =
  if m_predios == 0
    nothing
  else
    predio(p, l, h)
    rua_predios(p + vx(l + s), m_predios - 1, l, h, s)
  end
```

A função que modela cada edifício é a seguinte:

```
predio(p, l, h) =
  box(p, l, l, h)
```



```
malha_predios(xyz(0, 0, 0), 10, 10, 100, 400, 40)
```

As duas seguintes funções modelam uma destas cidades:

```
malha_predios(p, n_ruas, m_predios, l, h, s) =
  if n_ruas == 0
    nothing
  else
    rua_predios(p, m_predios, l, h, s)
    malha_predios(p + vy(l + s), n_ruas - 1, m_predios, l, h, s)
  end
```

```
rua_predios(p, m_predios, l, h, s) =
  if m_predios == 0
    nothing
  else
    predio(p, l, h)
    rua_predios(p + vx(l + s), m_predios - 1, l, h, s)
  end
```

A função que modela cada edifício é a seguinte:

```
predio(p, l, h) =
  box(p, l, l, h)
```

Exercício 4.2.1 Torne a cidade gerada pela função anterior mais realista através da incorporação da aleatoriedade na altura de cada edifício, tal como se apresenta na Figura 4.2. Dada a altura máxima de um edifício, a altura actual deverá ser um valor aleatório entre 10% da altura máxima e 100% da altura máxima.

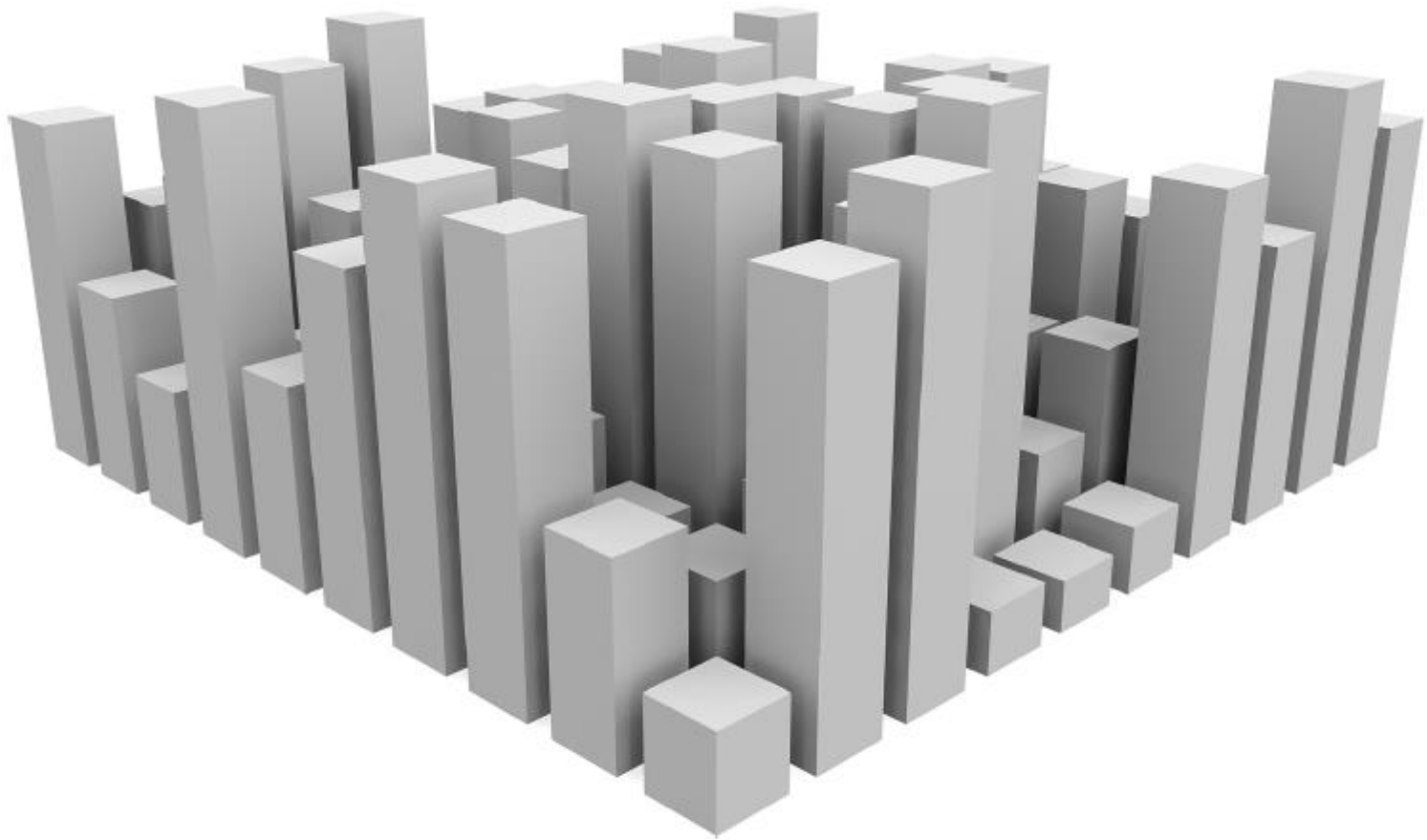


Figura 4.2: Uma urbanização em malha ortogonal com cem edifícios com alturas aleatórias.

EXERCÍCIO 4.2.1

```
malha_predios(p, n_ruas, m_predios, l, h, s) =  
  if n_ruas == 0  
    nothing  
  else  
    rua_predios(p, m_predios, l, h, s)  
    malha_predios(p + vy(l + s), n_ruas - 1, m_predios, l, h, s)  
  end
```

```
rua_predios(p, m_predios, l, h, s) =  
  if m_predios == 0  
    nothing  
  else  
    predio(p, l, h)  
    rua_predios(p + vx(l + s), m_predios - 1, l, h, s)  
  end
```

```
predio(p, l, h) = box(p, l, l, h)
```

EXERCÍCIO 4.2.1

```
malha_predios(p, n_ruas, m_predios, l, h, s) =  
  if n_ruas == 0  
    nothing  
  else  
    rua_predios(p, m_predios, l, h, s)  
    malha_predios(p + vy(l + s), n_ruas - 1, m_predios, l, h, s)  
  end
```

```
rua_predios(p, m_predios, l, h, s) =  
  if m_predios == 0  
    nothing  
  else  
    predio(p, l, h)  
    rua_predios(p + vx(l + s), m_predios - 1, l, h, s)  
  end
```

```
predio(p, l, h_max) = box(p, l, l, h_max * random_range(0.1, 1.0))
```

EXERCÍCIO 4.2.1

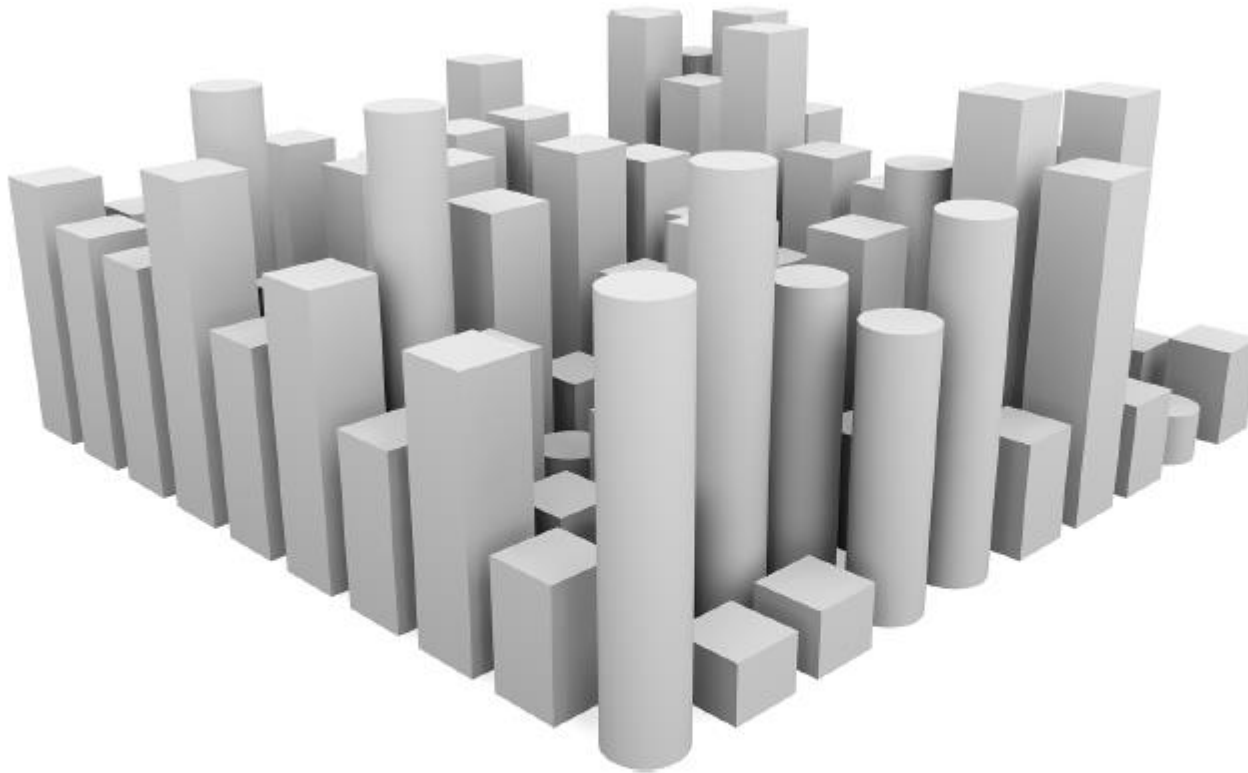
```
malha_predios(p, n_ruas, m_predios, l, h, s) =  
  if n_ruas == 0  
    nothing  
  else  
    rua_predios(p, m_predios, l, h, s)  
    malha_predios(p + vy(l + s), n_ruas - 1, m_predios, l, h, s)  
  end
```

```
rua_predios(p, m_predios, l, h, s) =  
  if m_predios == 0  
    nothing  
  else  
    predio(p, l, h)  
    rua_predios(p + vx(l + s), m_predios - 1, l, h, s)  
  end
```

```
predio(p, l, h_max) = box(p, l, l, h_max * random_range(0.1, 1.0))
```

```
malha_predios(xyz(0, 0, 0), 10, 10, 100, 400, 40)
```


Exercício 4.2.2 As urbanizações produzidas pelas funções anteriores não apresentam variabilidade suficiente pois os edifícios têm todos a mesma forma. Para melhorar a qualidade estética da urbanização pretende-se empregar diferentes funções para a construção de diferentes tipos de edifícios: a função `predio0` deverá construir um paralelepípedo de altura aleatória tal como anteriormente e a função `predio1` deverá construir uma torre cilíndrica de altura aleatória e contida no espaço de um quarteirão. Defina estas duas funções.

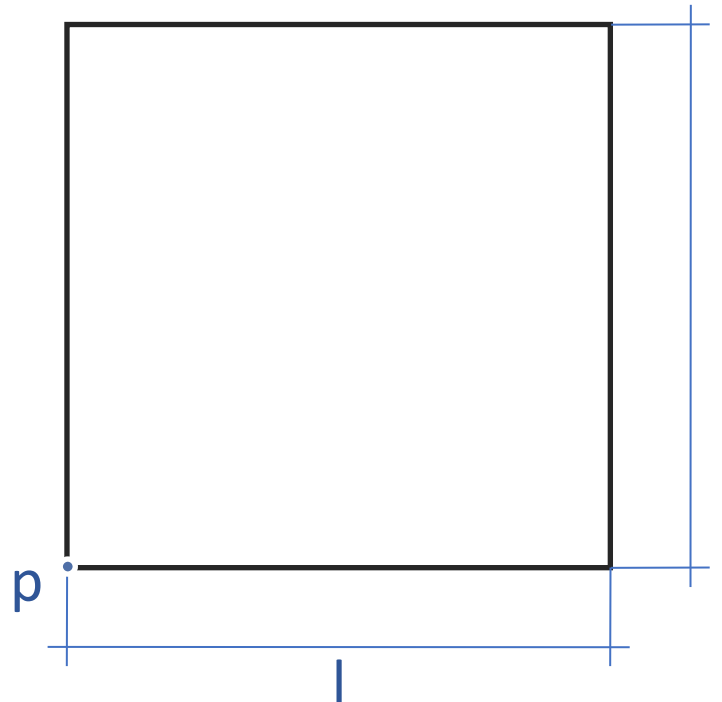


Exercício 4.2.4 As cidades criadas nos exercícios anteriores apenas permitem dois tipos de edifícios: torres paralelepipedicas ou torres cilíndricas.

EXERCÍCIO 4.2.2

`predio0(p, l, h_max) = box(p, l, l, h_max * random_range(0.1, 1.0))`

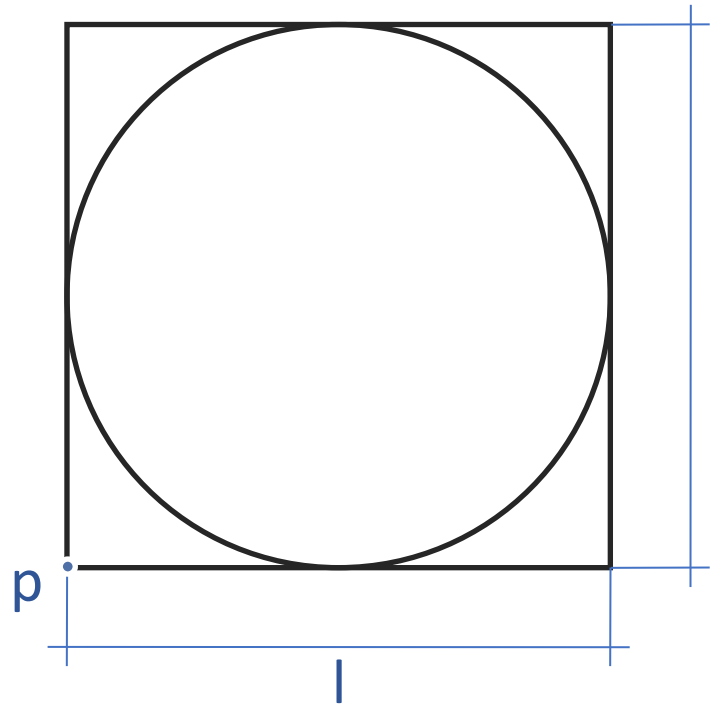
`predio1(p, l, h_max) = ?`



EXERCÍCIO 4.2.2

```
predio0(p, l, h_max) = box(p, l, l, h_max * random_range(0.1, 1.0))
```

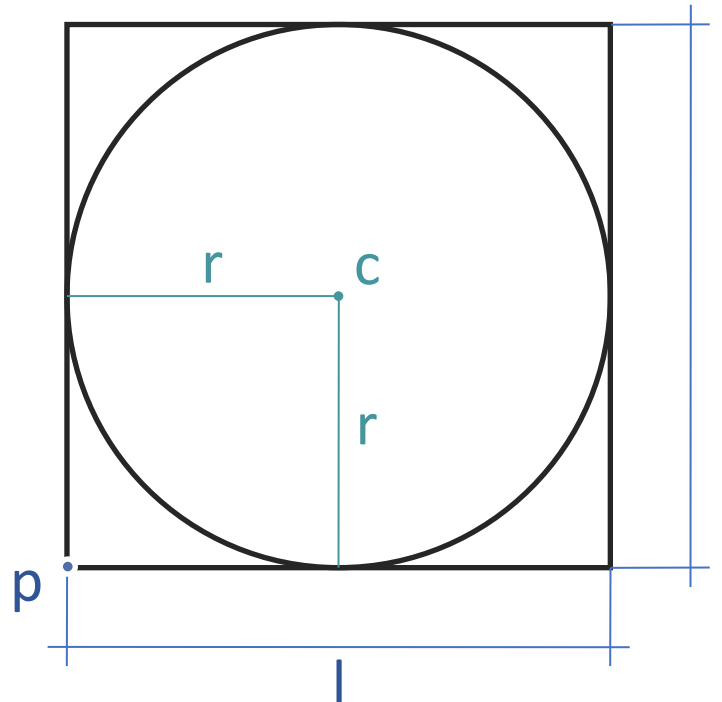
```
predio1(p, l, h_max) = cylinder(centro, raio, altura)
```



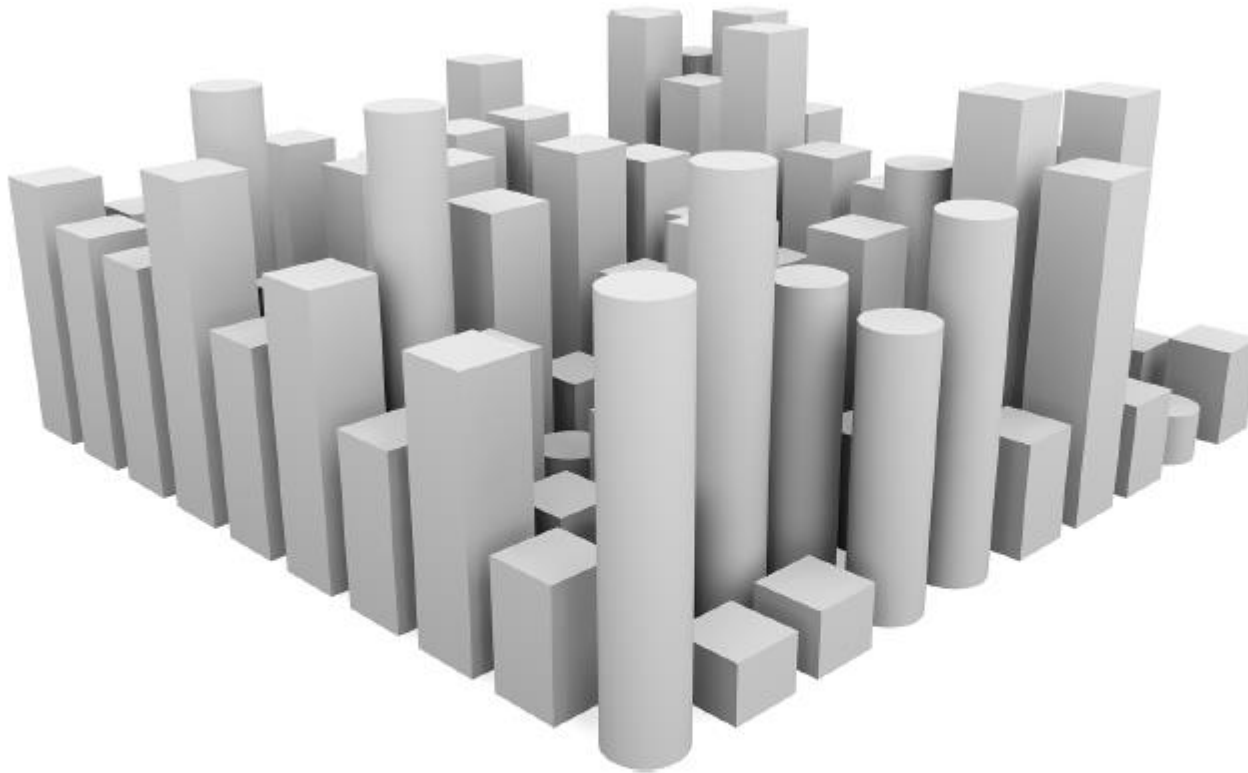
EXERCÍCIO 4.2.2

```
predio0(p, l, h_max) = box(p, l, l, h_max * random_range(0.1, 1.0))
```

```
predio1(p, l, h_max) = cylinder(centro, raio, altura)
```



Exercício 4.2.3 Pretende-se que use as duas funções `predio0` e `predio1` definidas no exercício anterior para a redefinição da função `predio` de modo a que esta, aleatoriamente, invoque uma ou outra de modo a construir prédios diferentes. A urbanização resultante deverá ser constituída aproximadamente por 20% de torres circulares e 80% de prédios rectangulares, tal como é exemplificado na imagem seguinte:

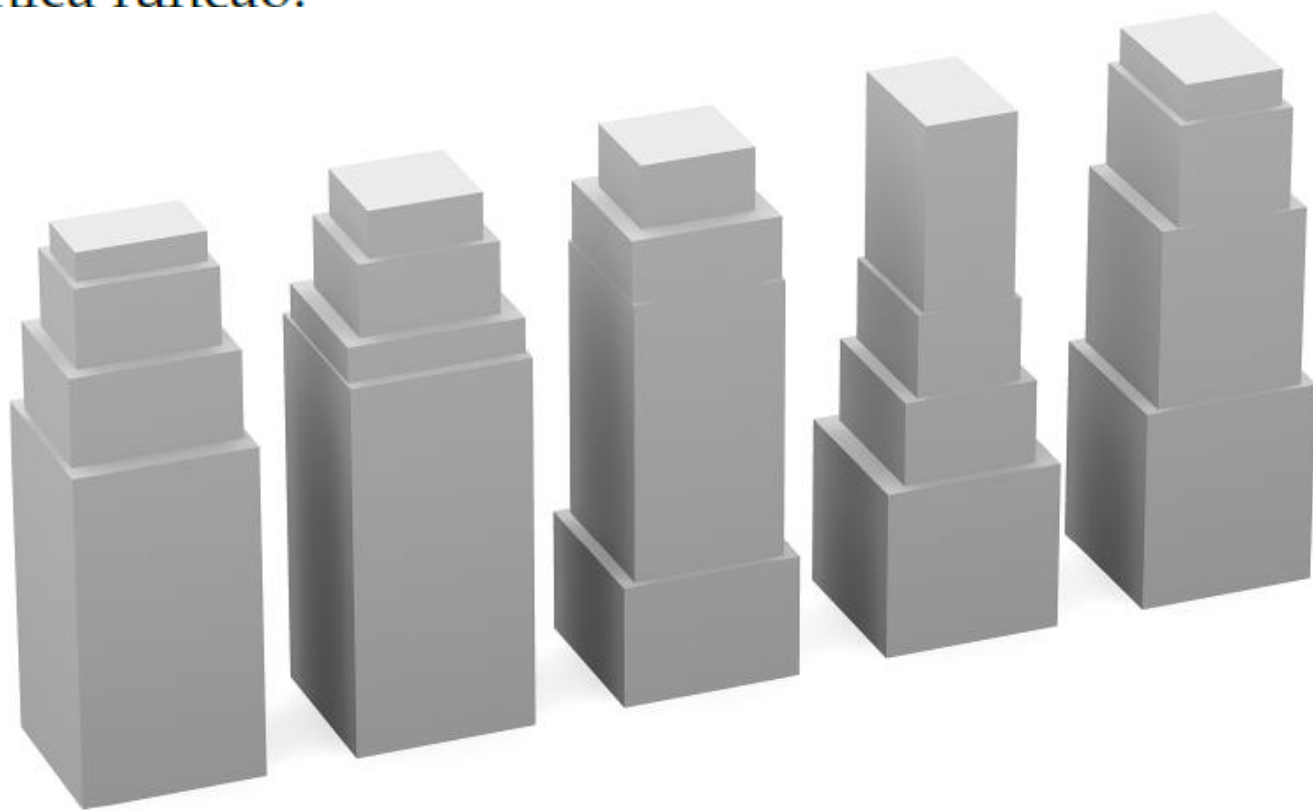


Exercício 4.2.4 As cidades criadas nos exercícios anteriores apenas permitem dois tipos de edifícios: torres paralelepípedicas ou torres cilíndricas.

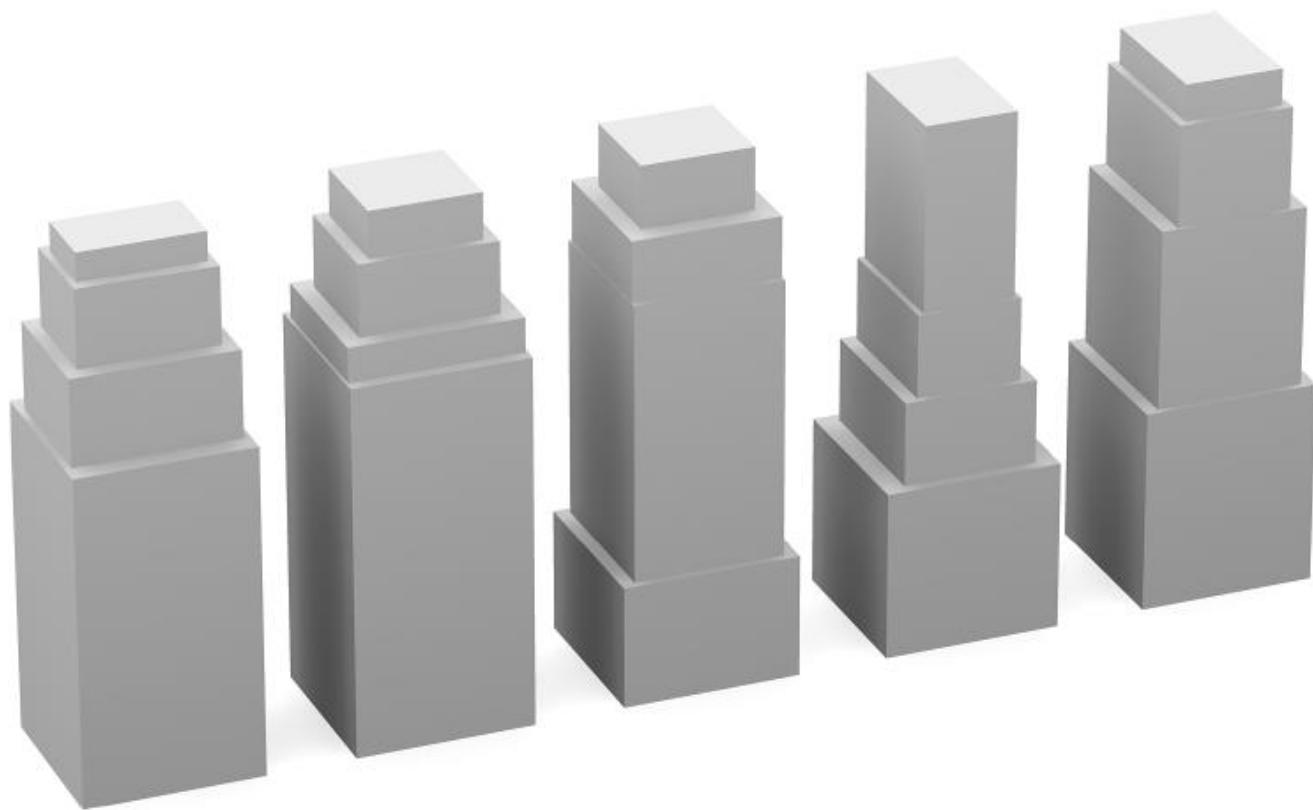
Exercício 4.2.4 As cidades criadas nos exercícios anteriores apenas permitem dois tipos de edifícios: torres paralelepípedicas ou torres cilíndricas.

Contudo, quando observamos uma cidade real, verificamos que existem prédios com muitas outras formas pelo que, para aumentarmos o realismo das nossas simulações, teremos de implementar um vasto número de funções, cada uma capaz de construir um edifício diferente.

Felizmente, uma observação atenta mostra que, na verdade, muitos dos prédios seguem um padrão e podem ser modelados por paralelepípedos sobrepostos com dimensões aleatórias mas sempre sucessivamente mais pequenos em função da altura, algo que podemos facilmente implementar com uma única função.



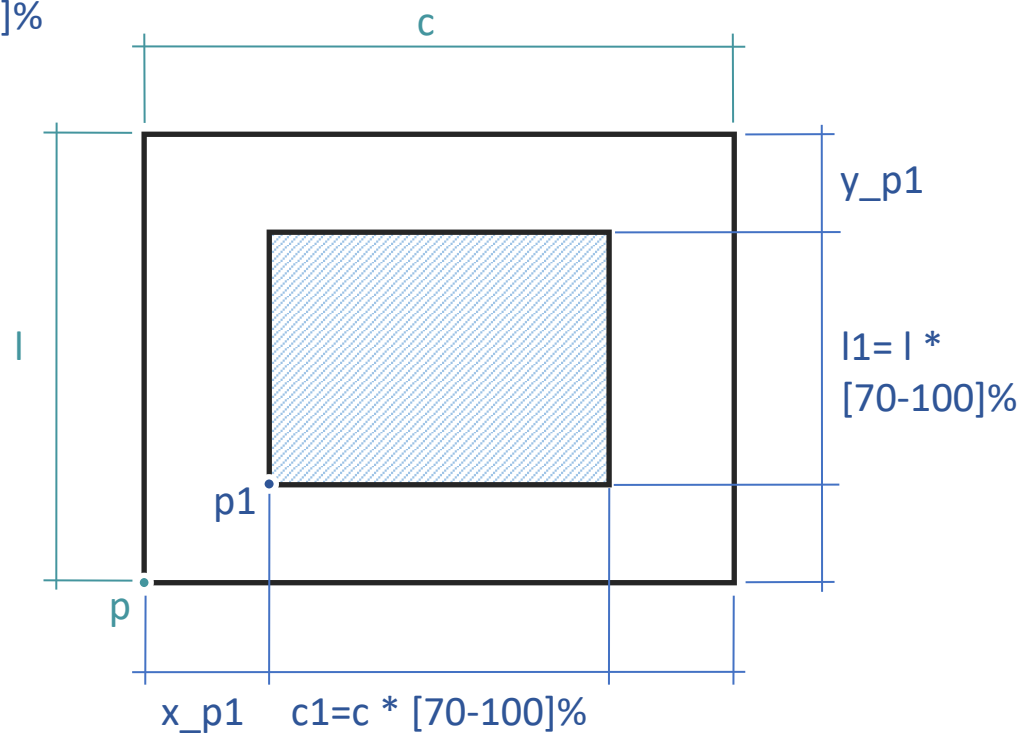
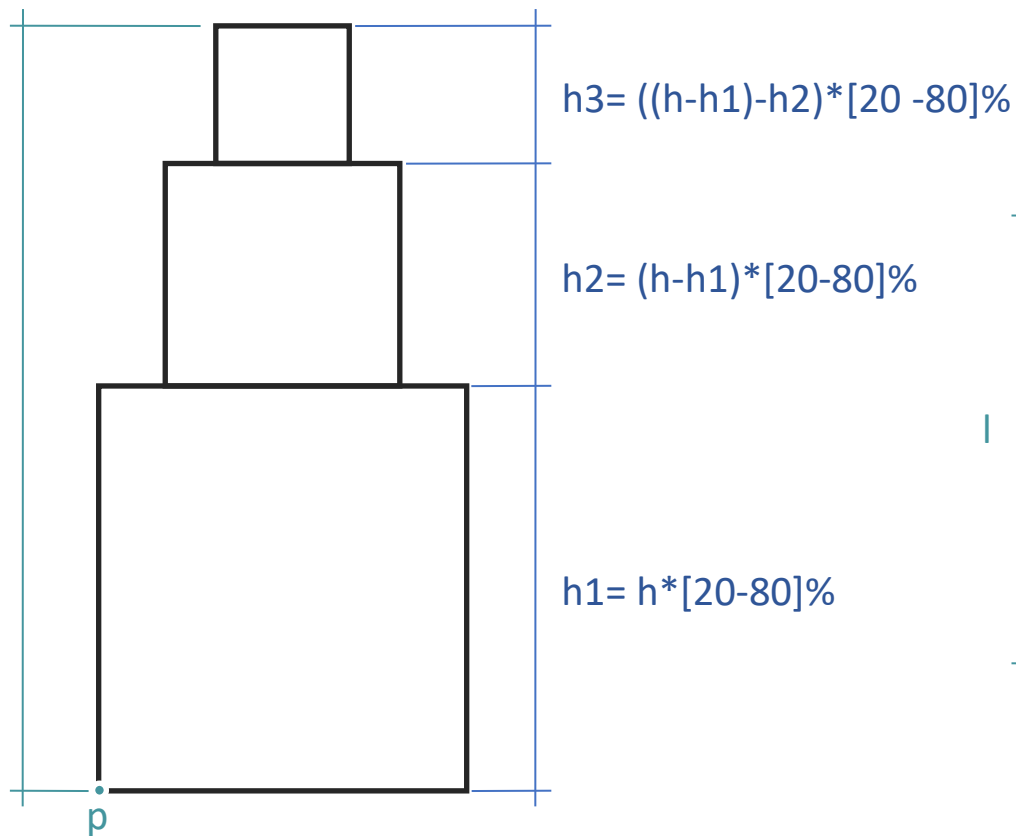
Considere que este modelo de edifício é parameterizado pelo número de “blocos” sobrepostos pretendidos, pelas coordenadas do primeiro bloco e pelo comprimento, largura e altura do edifício. O bloco de base tem exatamente o comprimento e largura especificados mas a sua altura deverá estar entre 20% e 80% da altura total do edifício. Os blocos seguintes estão centrados sobre o bloco imediatamente abaixo e possuem um comprimento e largura entre 70% e 100% dos parâmetros correspondentes desse bloco. A altura destes blocos deverá ser entre 20% e 80% da altura restante do edifício. A imagem seguinte mostra alguns exemplos deste modelo de edifícios:



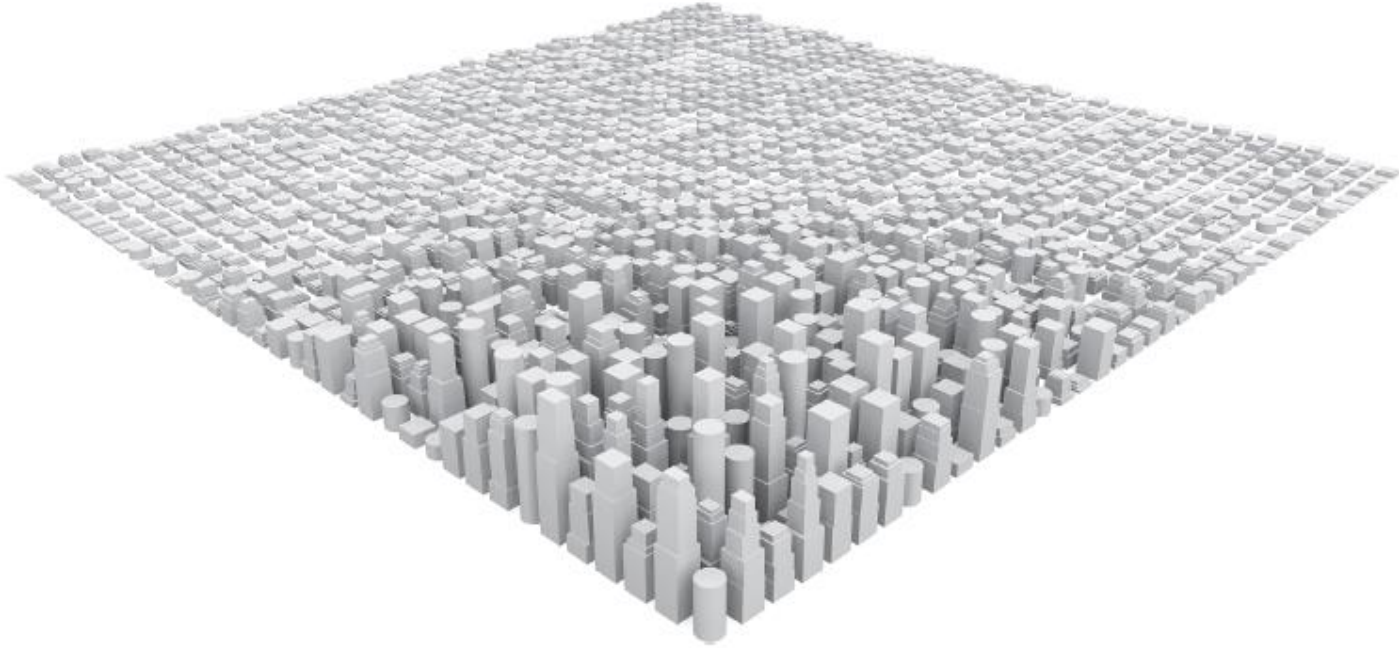
Com base nesta especificação, defina a função `predio_blocos` e use-a para redefinir a função `predio0` de modo a que sejam gerados prédios com um número aleatório de blocos sobrepostos. Com esta redefinição, a função `malha_predios` deverá ser capaz de gerar urbanizações semelhantes à imagem seguinte, em que se empregou para cada prédio um número de blocos entre 1 e 5:



EXERCÍCIO 4.2.4



Exercício 4.2.5 Em geral, as cidades possuem um núcleo de edifícios relativamente altos no centro e, à medida que nos afastamos para a periferia, a altura tende a diminuir, sendo este fenómeno perfeitamente visível na figura seguinte:



A variação da altura dos edifícios pode ser modelada por diversas funções matemáticas mas, neste exercício, pretende-se que empregue uma distribuição Gaussiana bi-dimensional dada por

$$f(x, y) = e^{-\left(\left(\frac{x-x_0}{\sigma_x}\right)^2 + \left(\frac{y-y_0}{\sigma_y}\right)^2\right)}$$

em que f é o factor de ponderação da altura, (x_0, y_0) são as coordenadas do ponto mais alto da superfície Gaussiana e σ_x e σ_y são os factores que afectam o alargamento bi-dimensional dessa superfície. Para simplificar, assuma que o centro da cidade fica nas coordenadas $(0, 0)$ e que $\sigma_x = \sigma_y = 25l$, sendo l a largura do edifício. Incorpore esta distribuição no processo de construção de cidades para produzir cidades mais realistas.