729G46 Informationsteknologi och programmering

Tema 5. Lektion inför temauppgift 5, Del 2

Jody Foo, jody.foo@liu.se

LINKÖPINGS UNIVERSITET

Temauppgift 5

- Mål

Placera ut fyrkanter i rader och kolumner

Inget överlapp får förekomma

Ingen fyrkant ska ligga utanför den givna ytan

Större fönster → fler kvadrater per rad/kolumn

Sluta när det inte får plats med fler kvadrater

- **Olika startpunkter**: från vänster/höger, uppifrån/nerifrån

- Utplaceringsalgoritmer

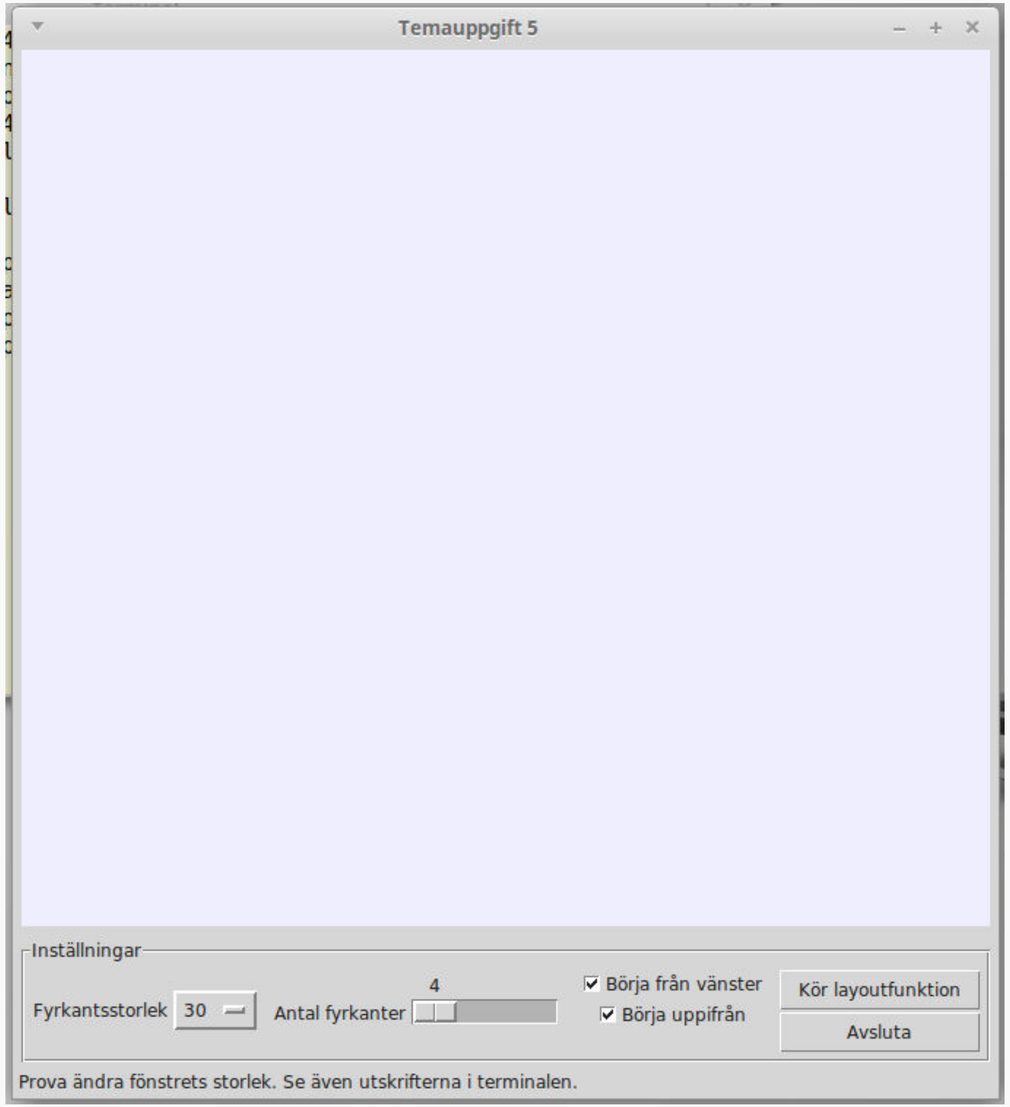
Ert gruppnummer avgör hur/i vilken ordning som fyrkanterna ska placeras ut

row layout

column layout

zig-zag layout (special: börjar alltid uppifrån)

Layouttestningsgränssnittet



Implementation

- **1 poäng**: alla kvadrater som ska placeras ut har samma storlek.

  Storleken kan variera mellan olika "layout-omgångar", men under en "layout-omgång" är alla kvadrater lika stora

- **3 poäng**: Slumpmässig storlek på kvadraterna för samma "layout-omgång"

Tre varianter

https://www.ida.liu.se/~729G46/tema5/temauppg5/#del2

Kod till temauppgiften

Filer

- `/courses/729G46/kursmaterial/temauppg5/del2`
- `tema5.py` - gränssnitt för att testa er layoutalgoritm
- `random-layout.py` - kodskelett

Kodskelett

- info på kurshemsidan
- .place()-metoden placerar kvadratens övre vänstra hörn

Demonstration

random-layout.py

Layoutfunktionen

- **Argument**

  en lista med fyrkanter (instanser av `tkinter.Label`)

  höjd och bredd på den tillgängliga ytan

  sanningsvärde: ska utplacering börja från vänster?

  sanningsvärde: ska utplacering börja uppifrån?

- Funktionen ska placera ut fyrkanterna med hjälp av geometrihanteraren `place` (dvs genom att ange x och y koordinater för det översta vänstra hörnet hos fyrkanten)

- `square.place(x=100, y=100)`

random-layout.py

```python
import tema5
import random


def random_layout(squares, frame_height, frame_width, start_left, start_top):
    """Placera ut kvadraterna i listan squares slumpmässigt.

    Argument:
    squares      -- Lista som innehåller tkinter.Label-objekt
    frame_height -- Höjden (int) på den Frame som kvadraterna ligger i
    frame_width  -- Bredden (int) på den Frame som kvadraterna ligger i
    start_left   -- Värdet True betyder att kvadraterna ska placeras ut
                    från vänster sida. Värdet False betyder att
                    kvadraterna ska placeras ut från höger.
    start_top    -- Värdet True betyder att kvadraterna ska börja placeras ut
                    uppifrån. Värdet False anger att kvadraterna ska börja
                    placeras ut från botten.
    """
    # Slumpa ut positioner för alla kvadrater utan att de hamnar utanför framen
    for square in squares:
        square_size = square.winfo_width()
        xpos = random.randint(0, frame_width - square_size)
        ypos = random.randint(0, frame_height - square_size)
        square.place(x=xpos, y=ypos)


if __name__ == "__main__":
    # layoutfunktionen som anropas av knappen "Kör layoutfunktion" skickas med
    # som ett funktionsobjekt när vi skapar en instans av LayoutTester (GUI:t)
    layout_tester = tema5.LayoutTester(random_layout)
```

random-layout.py

```
import tema5
import random


def my_layout(squares, frame_height, frame_width, start_left, start_top):
    """Placera ut kvadraterna i listan squares slumpmässigt.

    Argument:
    squares      -- Lista som innehåller tkinter.Label-objekt
    frame_height -- Höjden (int) på den Frame som kvadraterna ligger i
    frame_width  -- Bredden (int) på den Frame som kvadraterna ligger i
    start_left   -- Värdet True betyder att kvadraterna ska placeras ut
                    från vänster sida. Värdet False betyder att
                    kvadraterna ska placeras ut från höger.
    start_top    -- Värdet True betyder att kvadraterna ska börja placeras ut
                    uppifrån. Värdet False anger att kvadraterna ska börja
                    placeras ut från botten.
    """
    # initiala värden på xpos och ypos
    xpos = 0
    ypos = 0
    for square in squares:
        square.place(x=xpos, y=ypos)

        # räkna ut nästa x och y
        xpos = ...
        ypos = ...


if __name__ == "__main__":
    # layoutfunktionen som anropas av knappen "Kör layoutfunktion" skickas med
    # som ett funktionsobjekt när vi skapar en instans av LayoutTester (GUI:t)
    layout_tester = tema5.LayoutTester(random_layout)
```

Att tänka på

- en iteration i loopen per kvadrat; dvs square.place() bör bara anropas en gång per iteration.
- i koden för en iteration, bestäm er för:
  - om ni ändrar x/y för kvadraten ni håller på att placera; dvs ni **räknar ut x/y först och sen använder square.place()**
  - eller om ni använder tidigare uträknad x/y för kvadraten; dvs att ni **först använder square.place() och sen uppdaterar x/y inför utplacering av nästa kvadrat**

random-layout.py

```
import tema5
import random


def my_layout(squares, frame_height, frame_width, start_left, start_top):
    """Placera ut kvadraterna i listan squares slumpmässigt.

    Argument:
    squares      -- Lista som innehåller tkinter.Label-objekt
    frame_height -- Höjden (int) på den Frame som kvadraterna ligger i
    frame_width  -- Bredden (int) på den Frame som kvadraterna ligger i
    start_left   -- Värdet True betyder att kvadraterna ska placeras ut
                    från vänster sida. Värdet False betyder att
                    kvadraterna ska placeras ut från höger.
    start_top    -- Värdet True betyder att kvadraterna ska börja placeras ut
                    uppifrån. Värdet False anger att kvadraterna ska börja
                    placeras ut från botten.
    """

    # kod som kanske tilldelar ngn variabel var vi börjar eller liknande
    ...
    for square in squares:
        # räkna ut x och y för nuvarande square
        xpos = ...
        ypos = ...

        # placera ut nuvarande square
        square.place(x=xpos, y=ypos)


if __name__ == "__main__":
    # layoutfunktionen som anropas av knappen "Kör layoutfunktion" skickas med
    # som ett funktionsobjekt när vi skapar en instans av LayoutTester (GUI:t)
    layout_tester = tema5.LayoutTester(random_layout)
```

Att tänka på

- en iteration i loopen per kvadrat; dvs square.place() bör bara anropas en gång per iteration.
- i koden för en iteration, bestäm er för:
  - om ni ändrar x/y för kvadraten ni håller på att placera; dvs ni **räknar ut x/y först och sen använder square.place()**
  - eller om ni använder tidigare uträknad x/y för kvadraten; dvs att ni **först använder square.place() och sen uppdaterar x/y inför utplacering av nästa kvadrat**

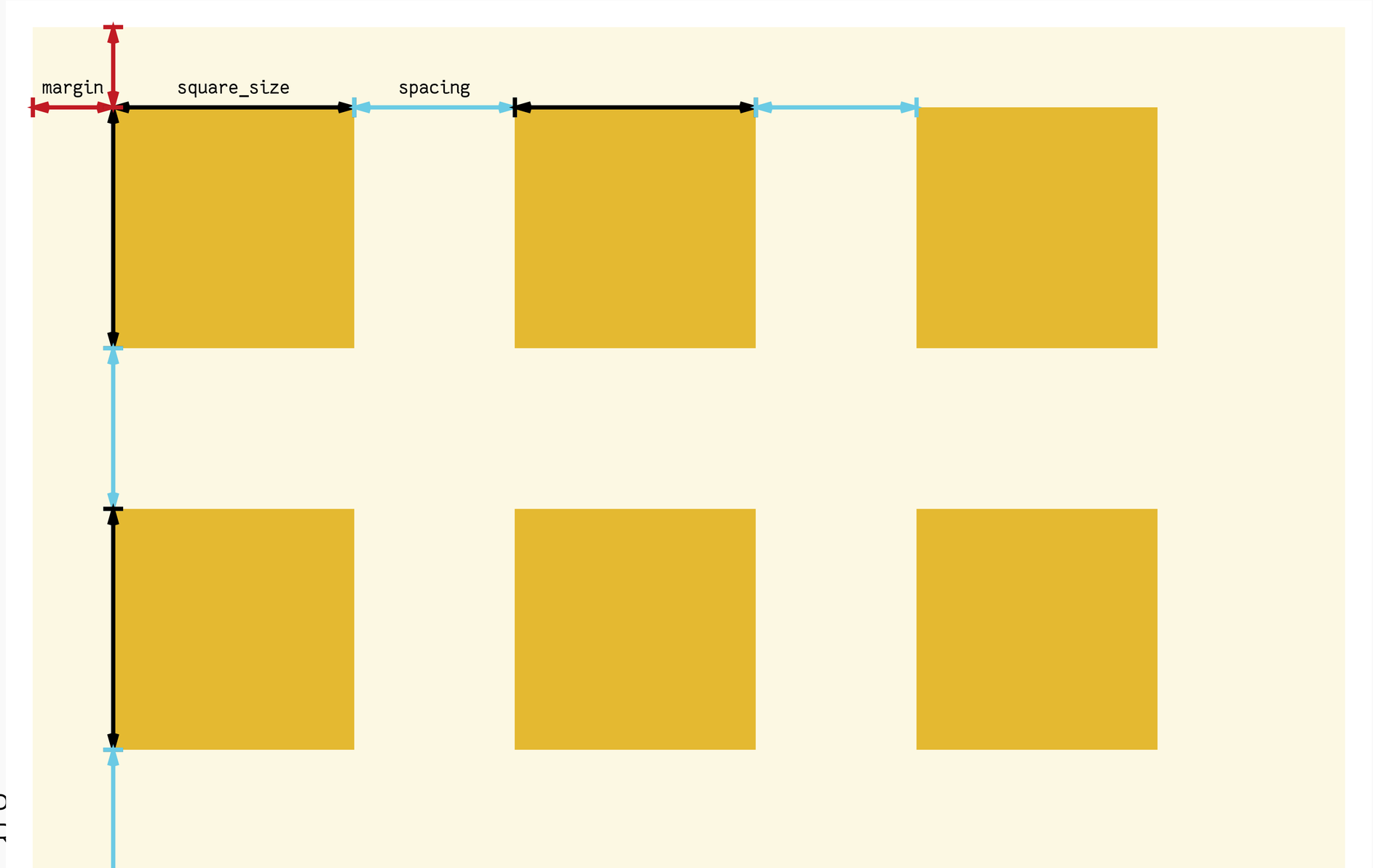
Skissa på er algoritm

Diskutera!

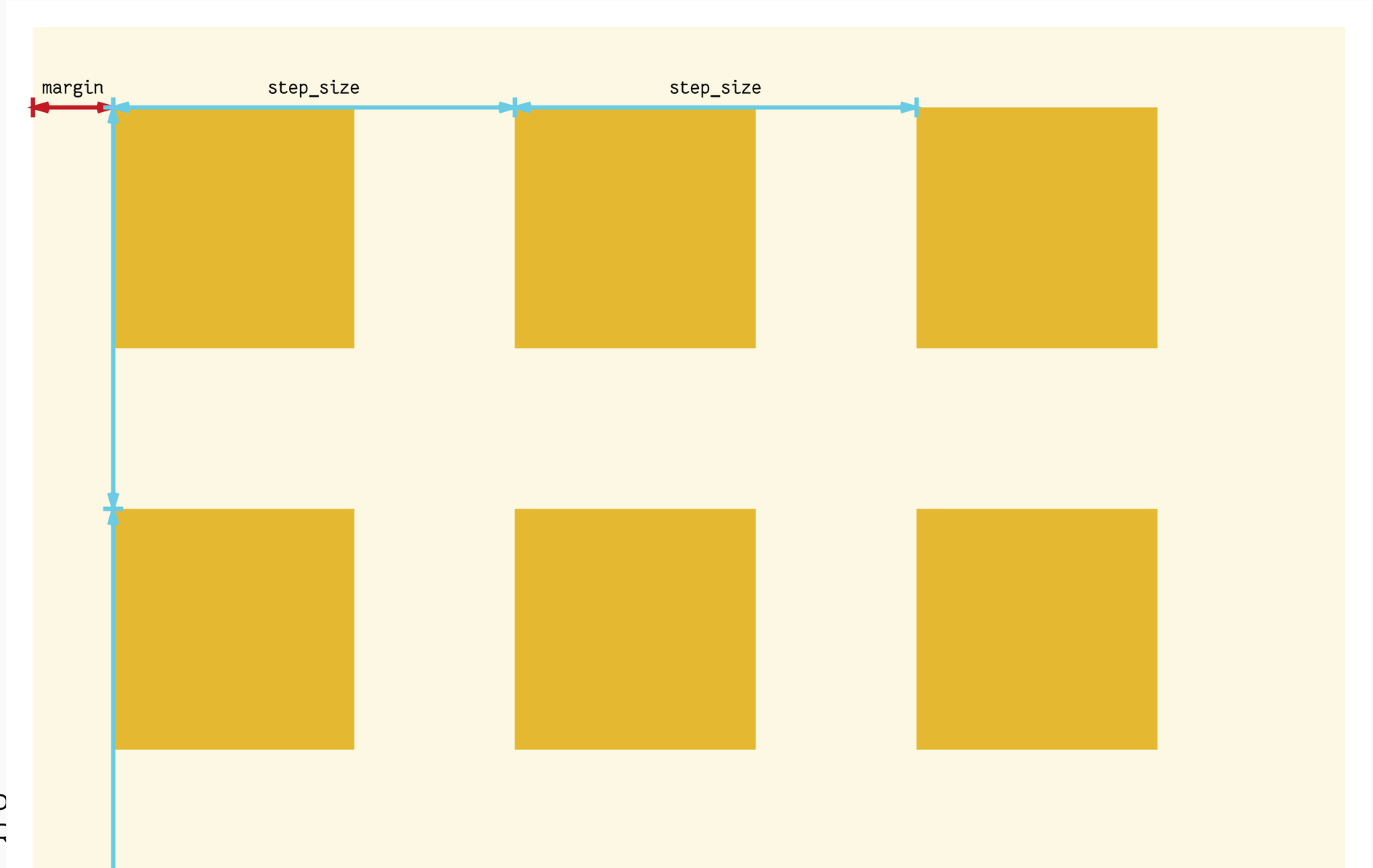
- Simulera utplacering av kvadrater med papper och penna "som människor"
- Fundera sedan vad som behöver göras för att räkna ut var en fyrkant ska placeras
- Kan ni identifiera delproblem?
- Vilka begrepp ("variabler") kan vara vettiga att introducera? Dvs vad behöver ni veta för att kunna räkna ut var en fyrkant ska placeras ut?

Exempel på angreppsätt

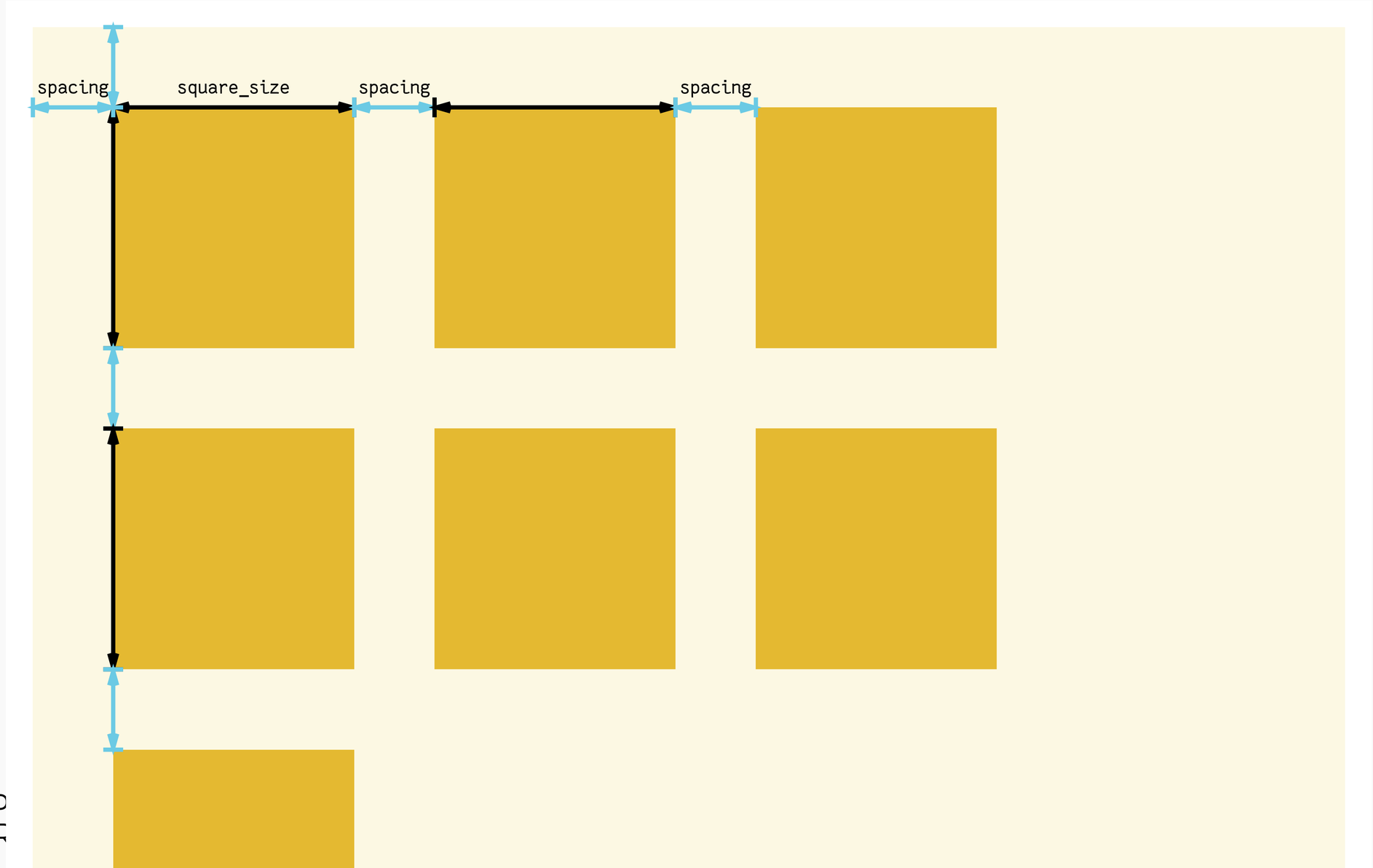
Ex. 1: Marginal + kvadrat + mellanrum



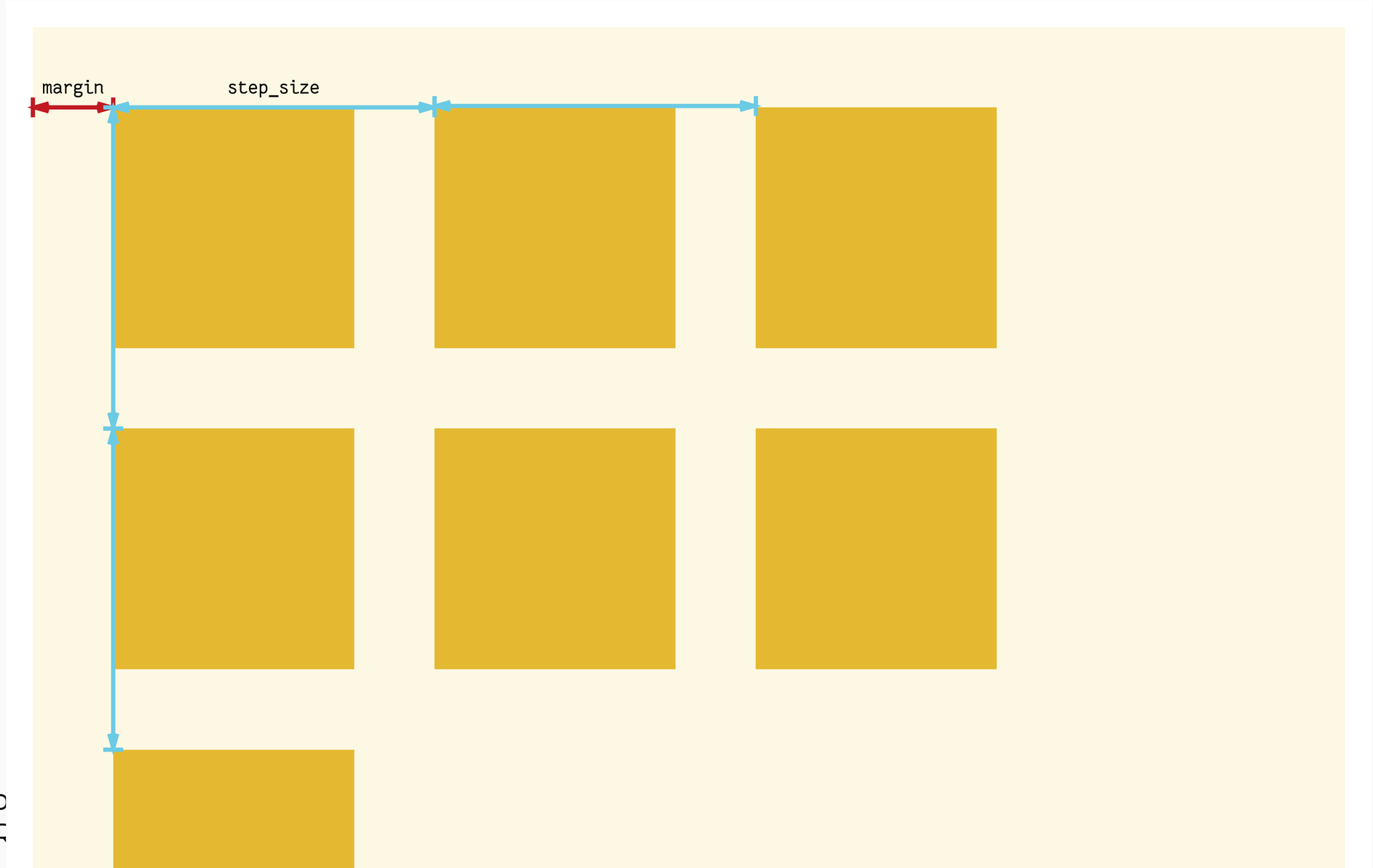
Ex. 2: Marginal + avstånd till nästa fyrkant



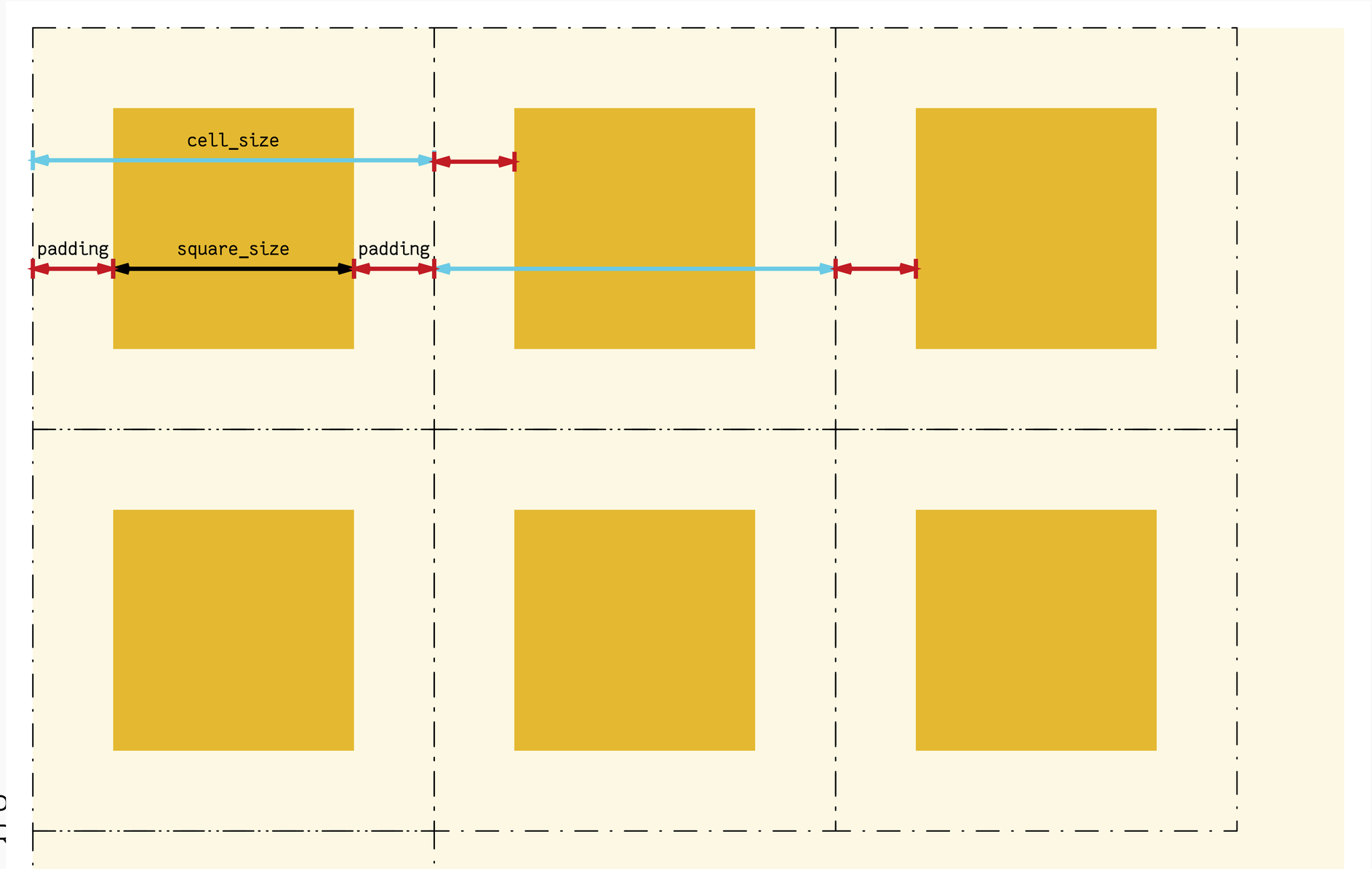
Ex. 3: Mellanrum + fyrkant (marginal har samma storlek som mellanrum)



Ex. 4: Marginal + avstånd till nästa (marginal har samma storlek som mellanrum)

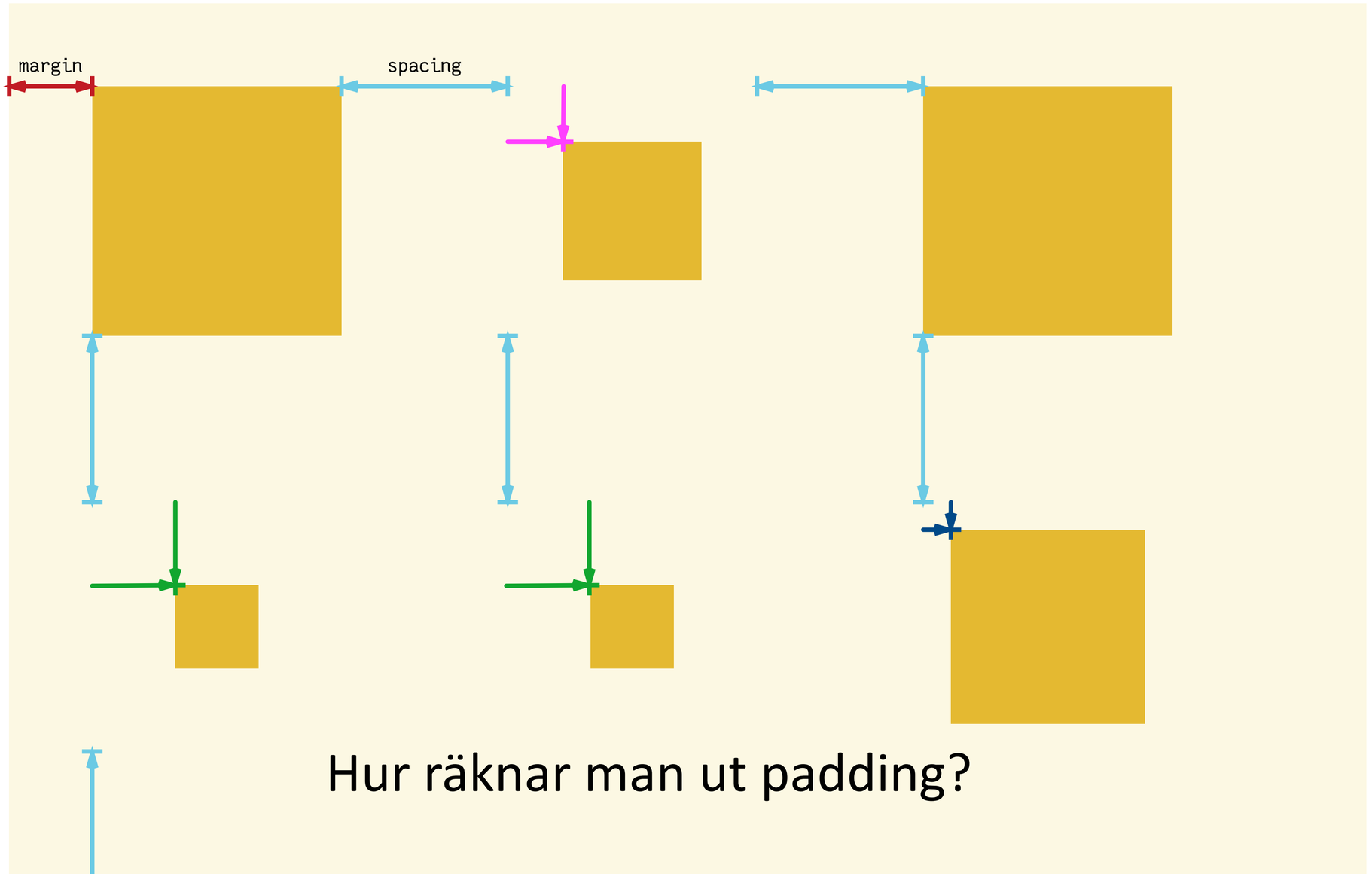


Ex. 5: Osynlig fyrkant runt fyrkanten, en cell. Inget mellanrum mellan celler, men "padding" inuti cellen.



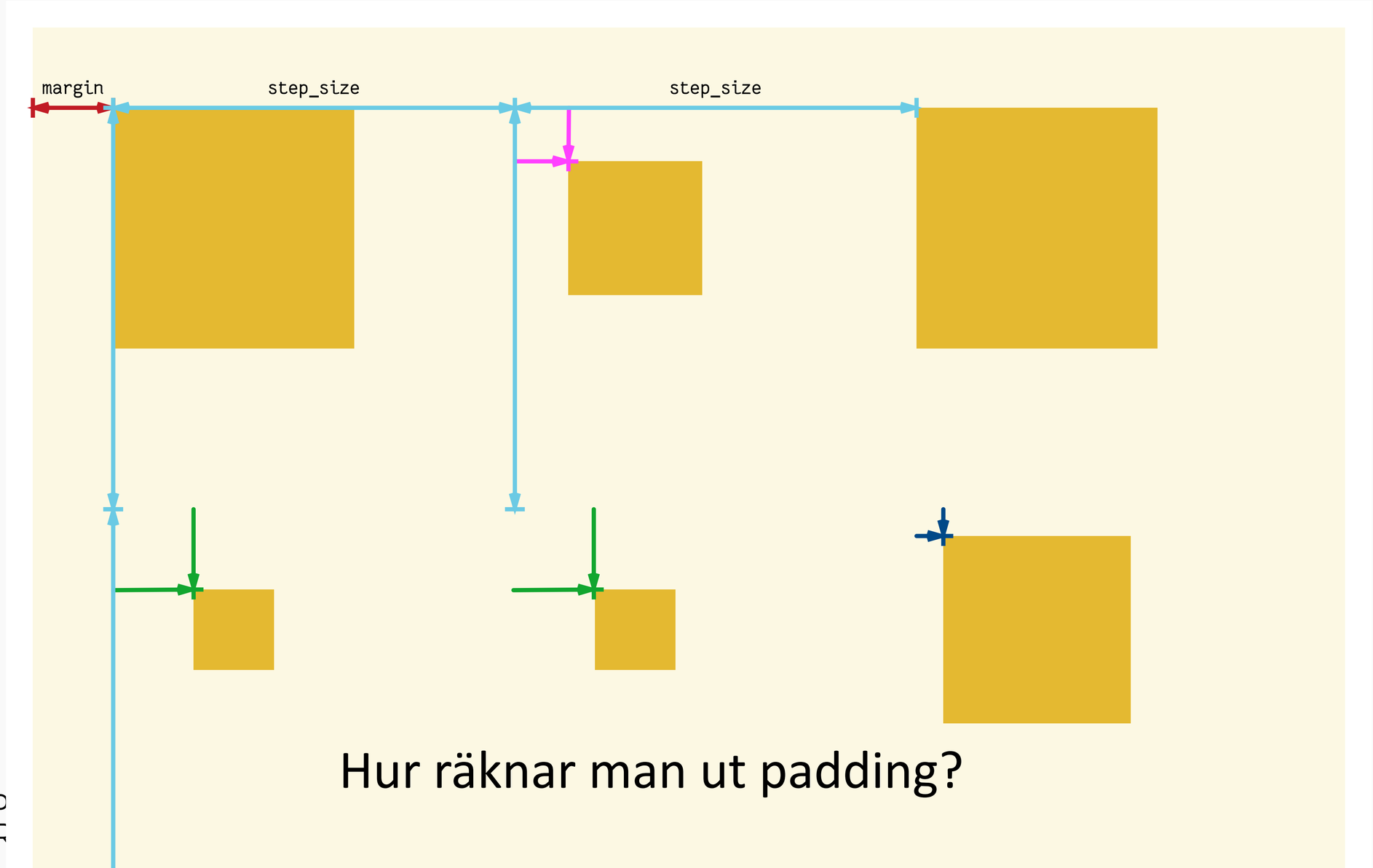
LINKÖP
UNIVER

Ex. 6: Slumpmässig storlek, marginal + mellanrum mellan celler + varierande padding

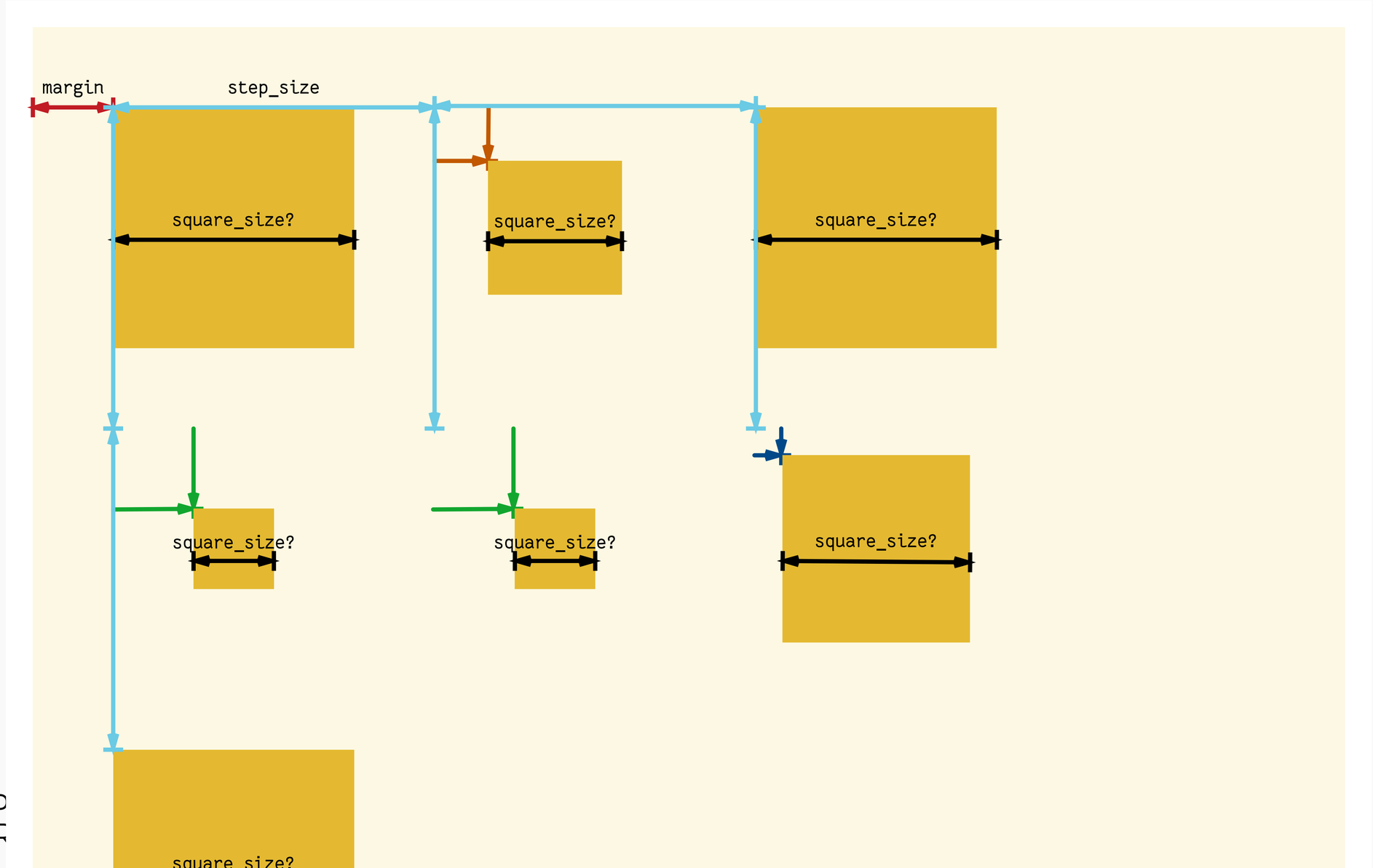


Hur räknar man ut padding?

Ex. 7a: Slumpmässig storlek, bara mellanrum mellan cellernas vänsterkant + varierande padding

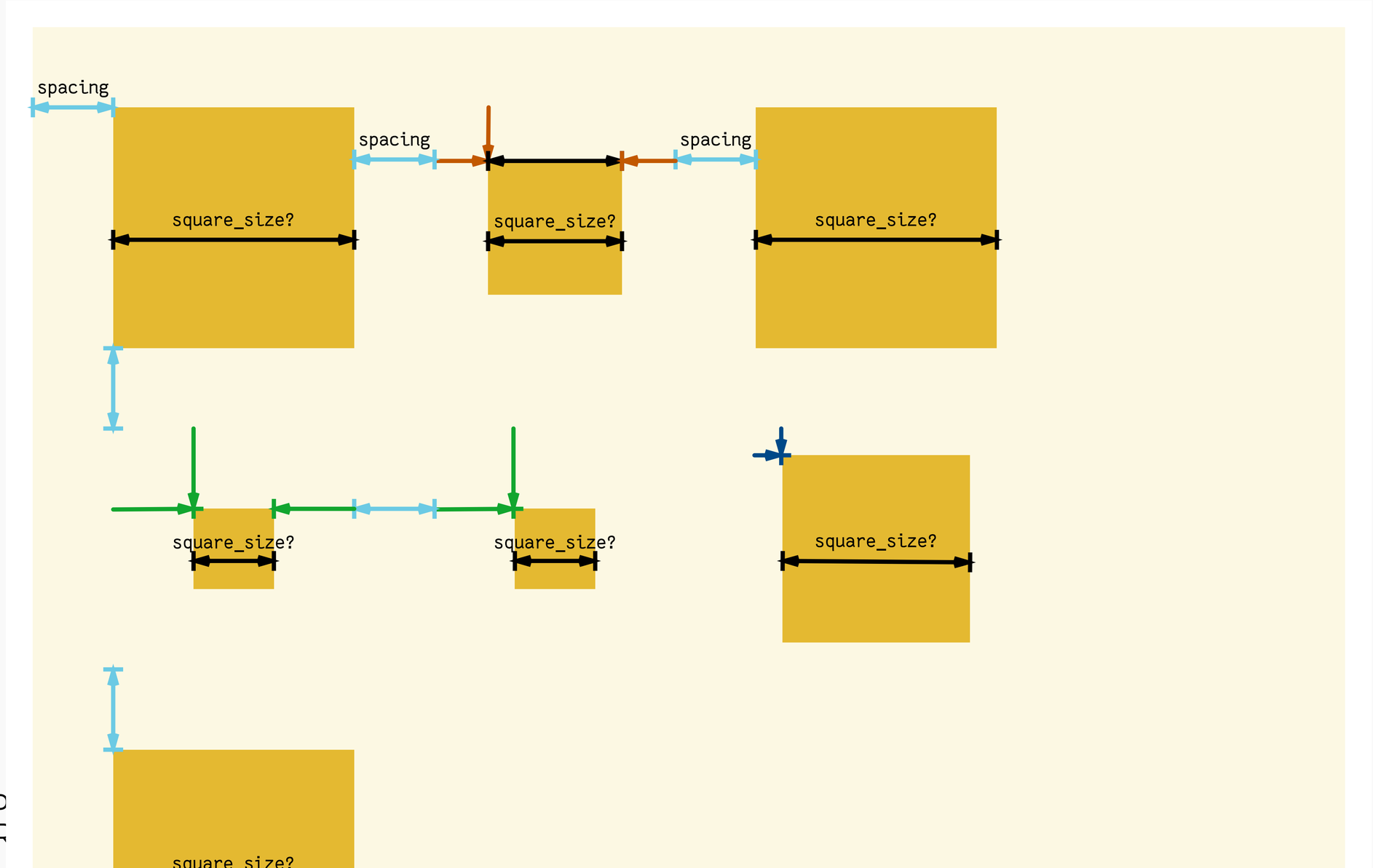


Ex. 7b: Slumpmässig storlek, bara mellanrum mellan rutornas vänsterkant + varierande padding



LINKÖP
UNIVER

Ex. 8: Slumpmässig storlek, mellanrum mellan celler + varierande padding



Ex. 9: Slumpmässig storlek, celler utan mellanrum, fast padding + varierande padding