

729G46 Informationsteknologi och programmering

Tema 5. Lektion inför temauppgift 5

Temauppgift 5

- Mål

Placera ut fyrkanter i rader och kolumner

Inget överlapp får förekomma

Ingen fyrkant ska ligga utanför den givna ytan

- Utplaceringsalgoritmer

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

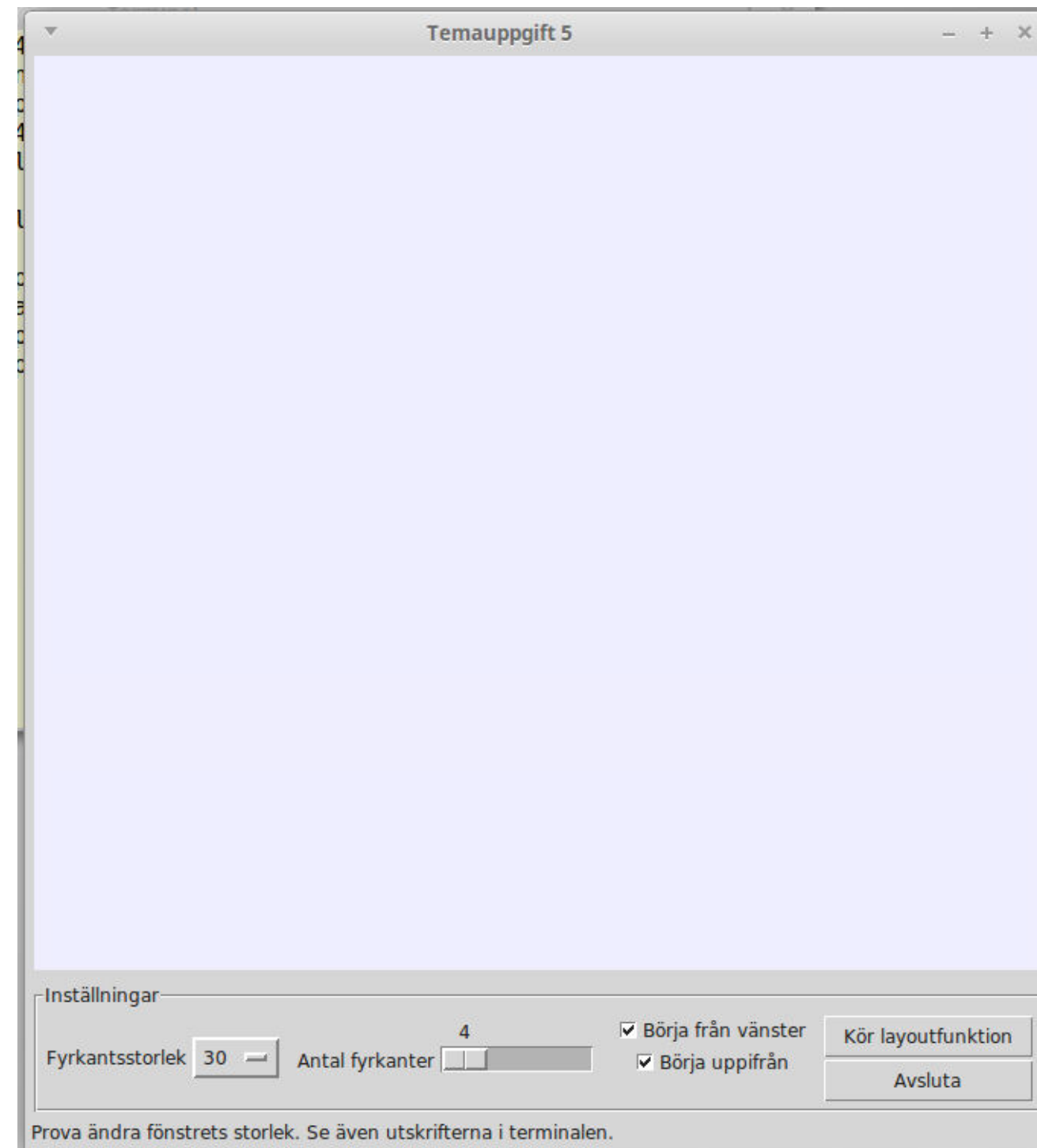
row layout

column layout

zig-zag layout

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

Layouttestningsgränssnittet



Implementation

- **G**: alla kvadrater som ska placeras ut har samma storlek.
Storleken kan variera mellan olika "layout-omgångar", men under ett och samma är alla kvadrater lika stora
- **G+**: Slumpmässig storlek på kvadraterna för samma "layout-omgång"

Tre varianter

<https://www.ida.liu.se/~729G46/tema5/temauppg5/#de12>

Kod- och kommentarskrav

Kod och kommentarskrav för G

- PEP8 + PEP257
- Inga variabelnamn som består av en bokstav.
- Kommentera alla satser som består av fler än en rad (if-t.ex. satser, loopar)

Kodkrav G+

- Inga hårdkodade värden. Variabler med bra namn gör det lättare läsa koden och att ändra på algoritmens beteende.
- Bryt ut funktioner - funktioner inte längre än 15 rader.
Lättare att läsa koden, att felsöka och underhålla koden.

Påminnelse om uppdaterat redovisningsförfarande

- Alla grupper ska ha tid att redovisa.
- ~10 minuter / grupp
- Ni visar att uppgifter fungerar som de ska och går igenom koden er kod på ett sätt som
 - täcker alla lärandemål för G/G+
 - visar att koden uppfyller kod/kommentarskrav för G/G+
- Handledaren ni redovisar för har tid för några enstaka klargörande frågor

Kod till temauppgiften

Filer

- </courses/729G46/kursmaterial/temauppg5/del2>
- [tema5.py](#) - gränssnitt för att testa er layoutalgoritm
- [random-layout.py](#) - kodskelett

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

```
import tema5
import random

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

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

    """
    # Slumpa ut positioner för alla fyrkanter 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)
```

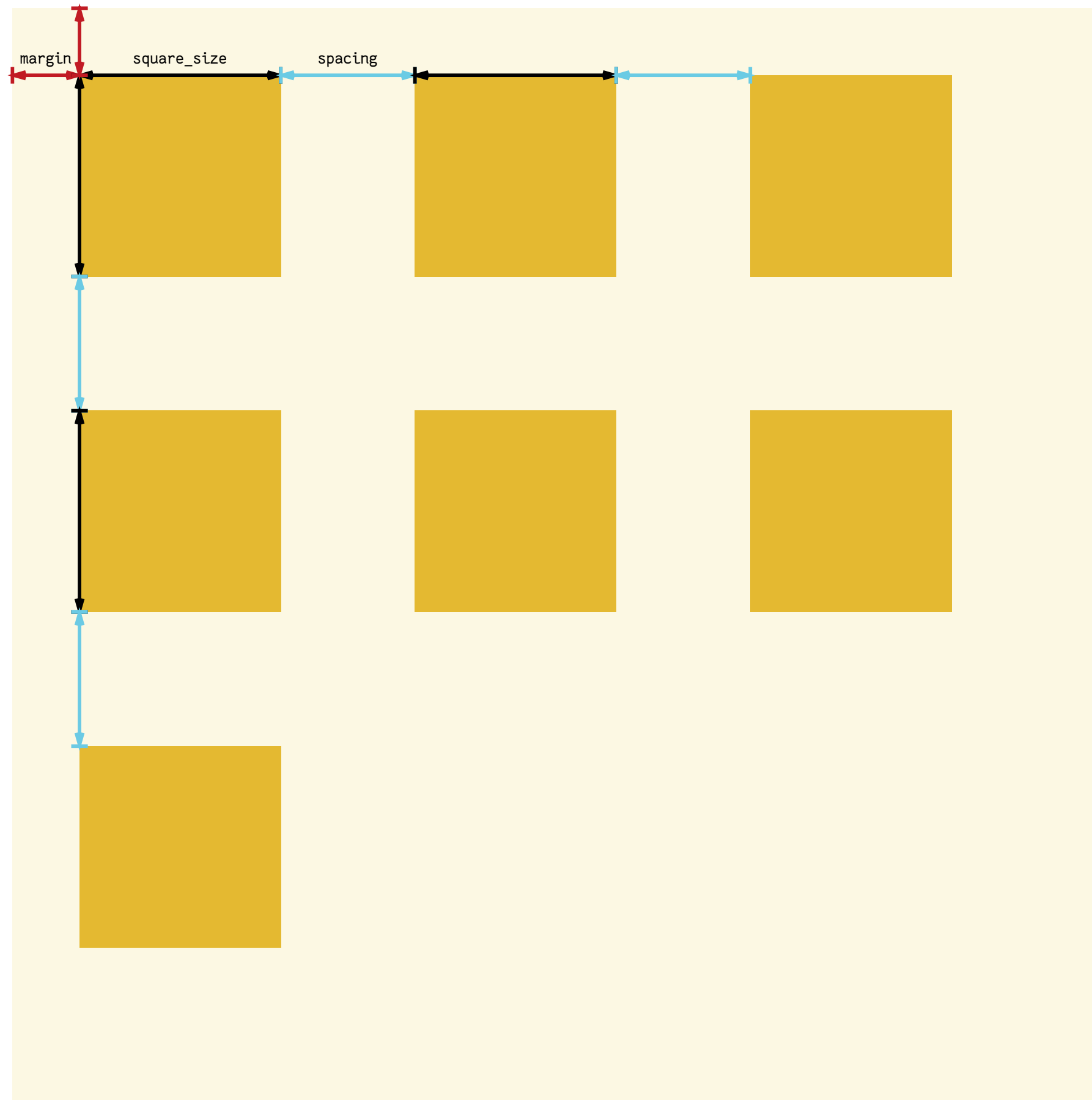
Skissa på er algoritm

Börja programmera genom att diskutera

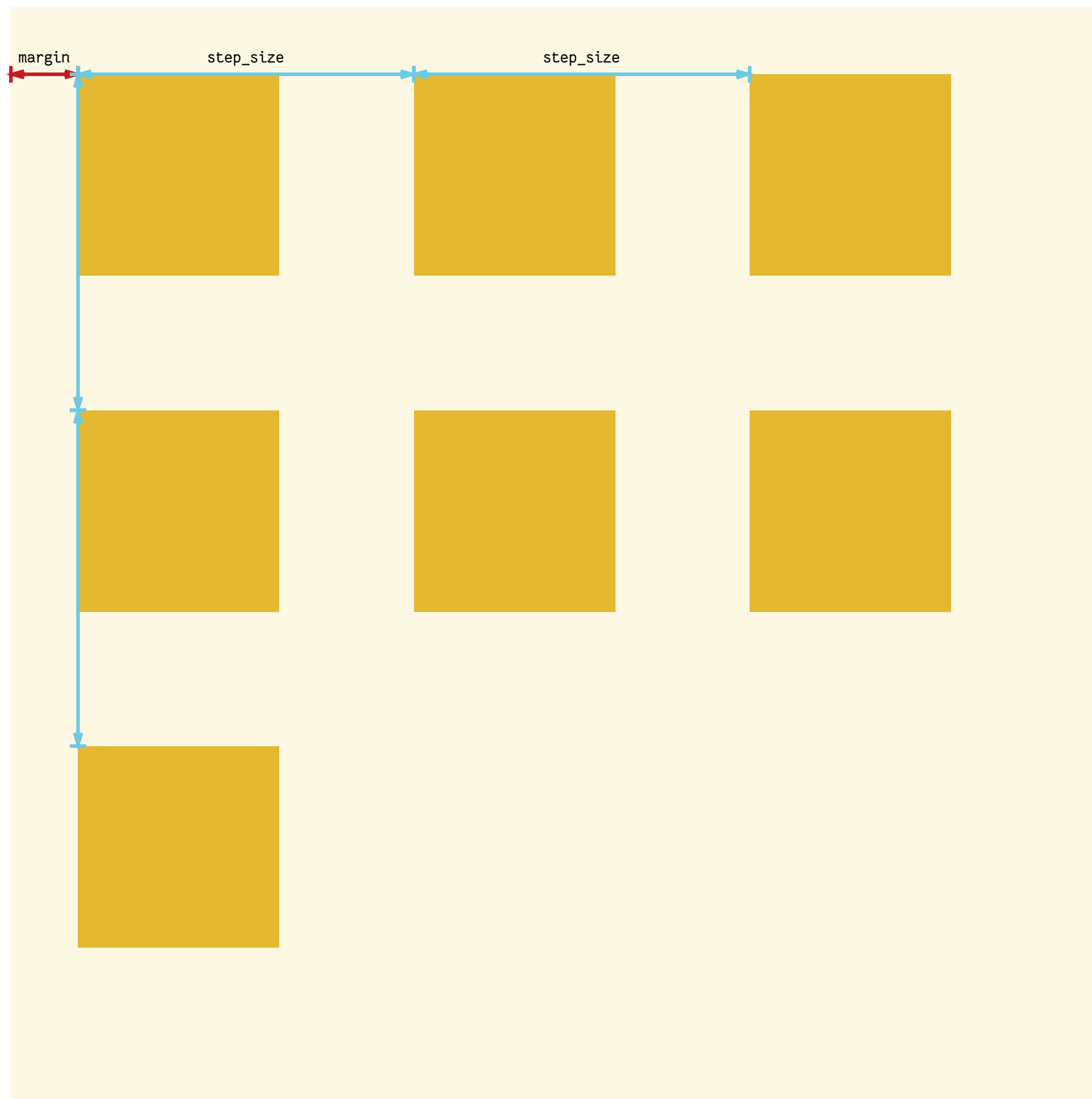
- Simulera utplacering av kvadrater med papper och penna "som människor"
- Fundera sedan vad som behöver göras för att räkna ut var en fyrkant ska placeras
- Kan ni identifiera delproblem?
- Vilka begrepp ("variabler") kan vara vettiga att introducera?
Dvs vad behöver ni veta för att kunna räkna ut var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

Exempel på angreppsätt

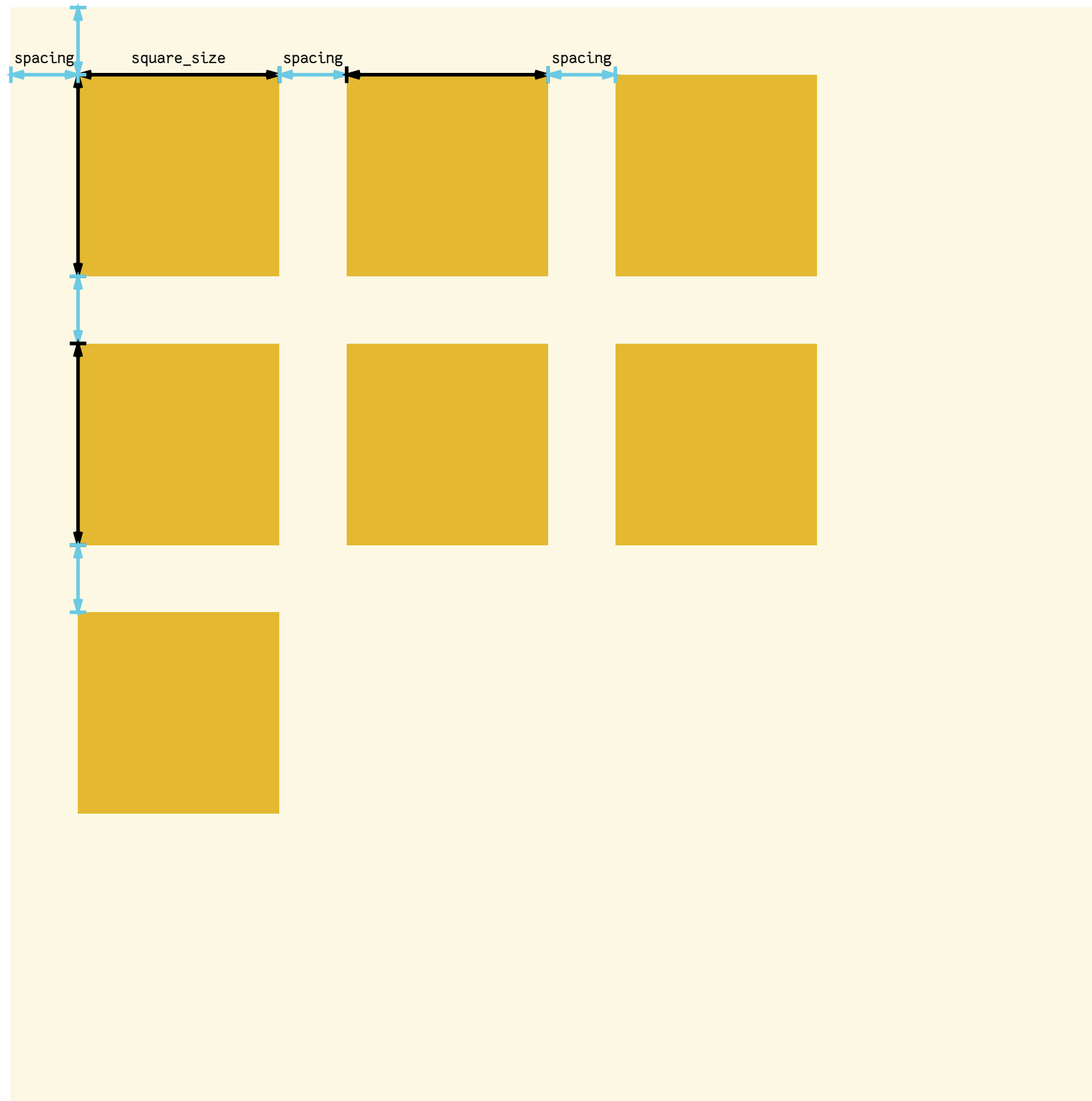
Ex. 1: Marginal + kvadrat + mellanrum



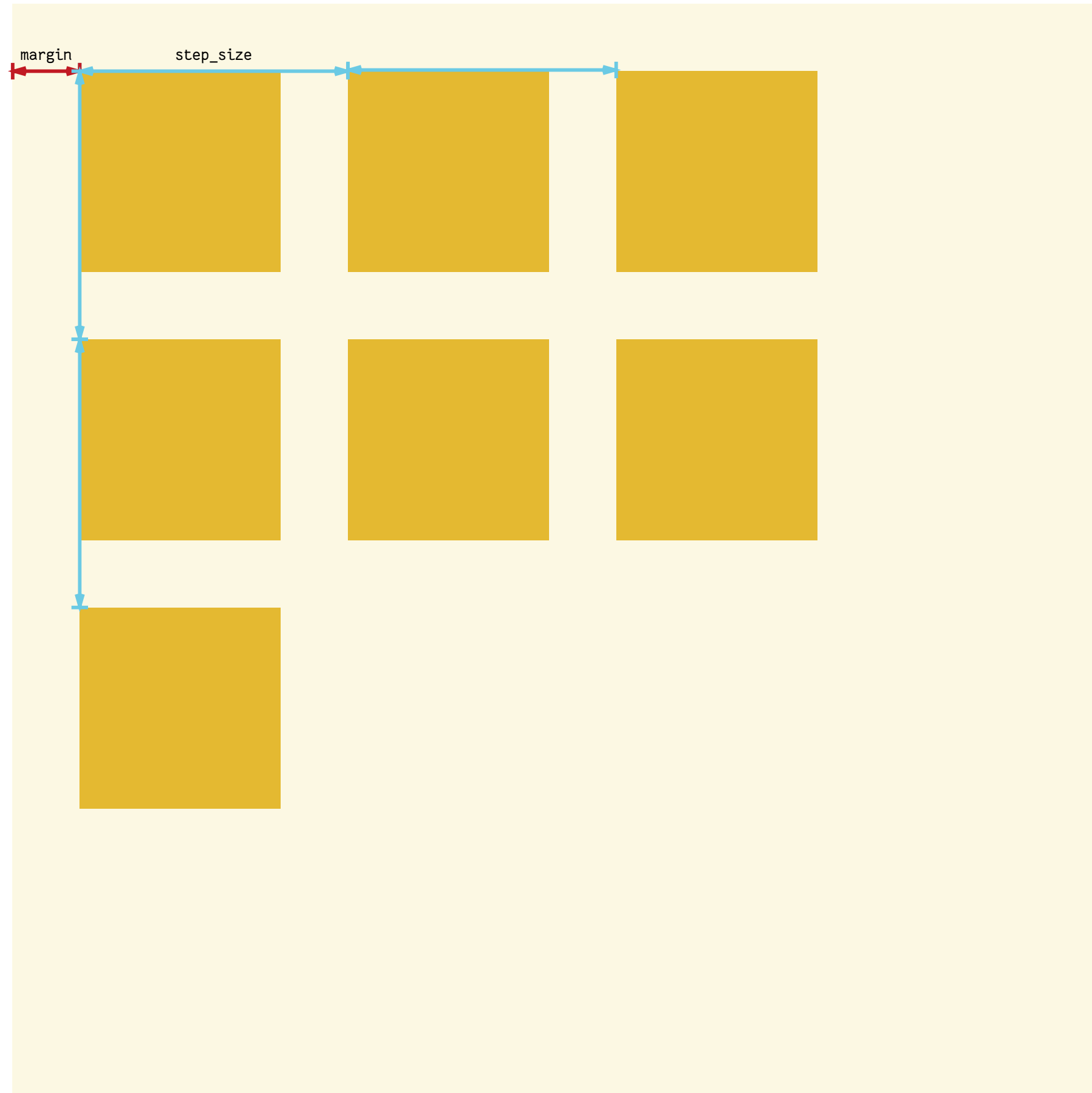
Ex. 2: Marginal + avstånd till nästa fyrkant



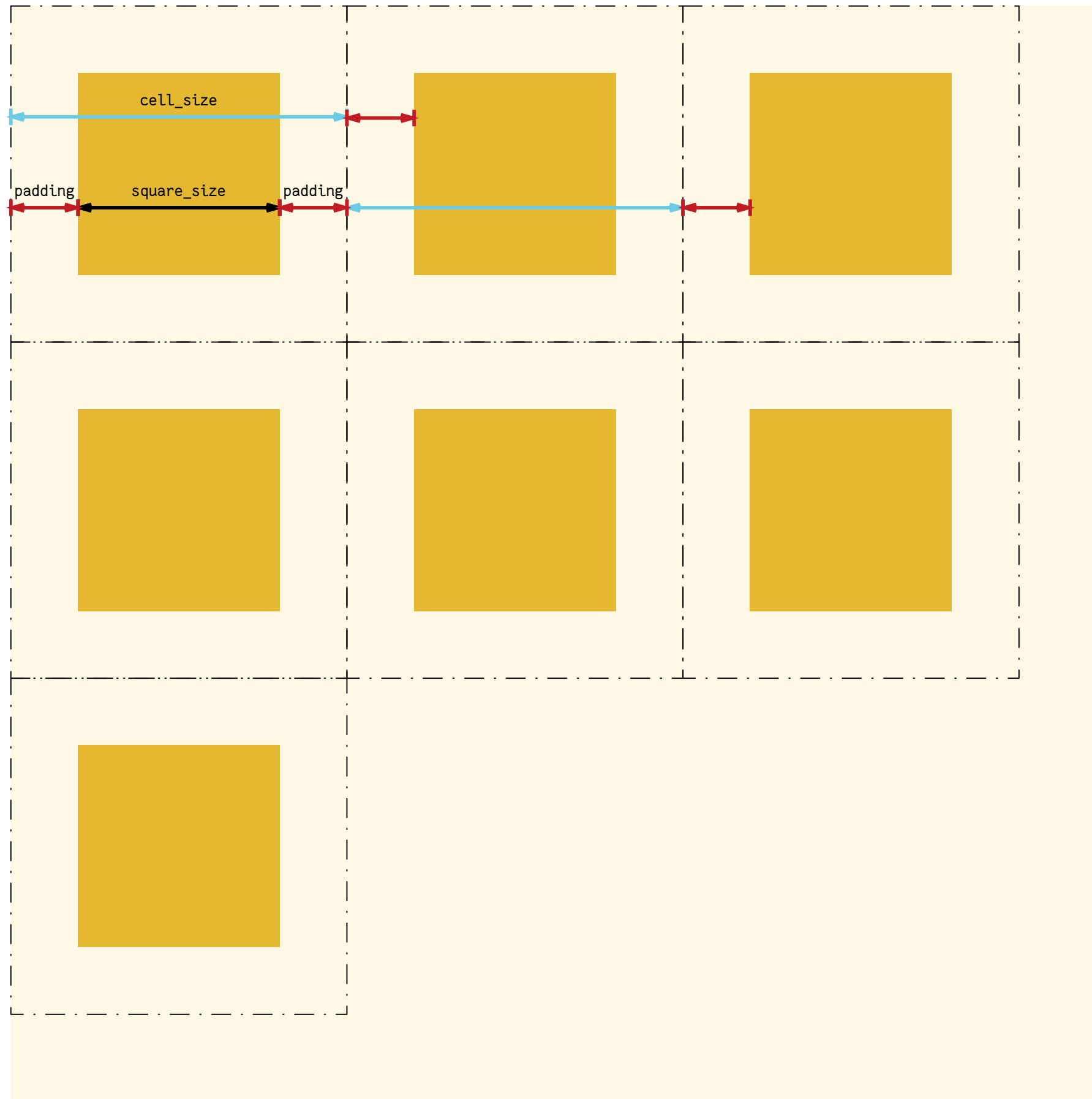
Ex. 3: Mellanrum + fyrkant (marginal har samma storlek som mellanrum)



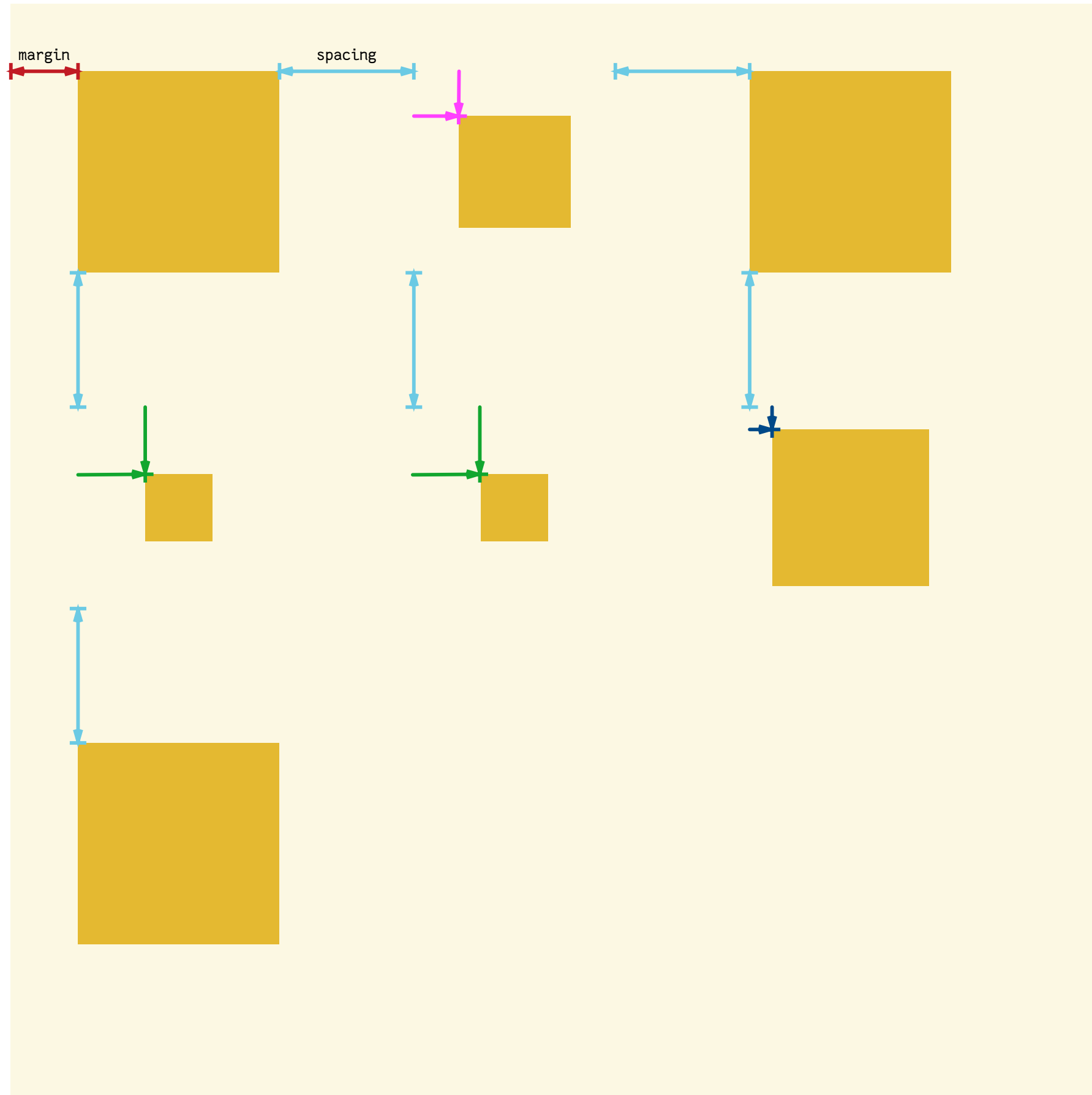
Ex. 4: Marginal + avstånd till nästa (marginal har samma storlek som mellanrum)



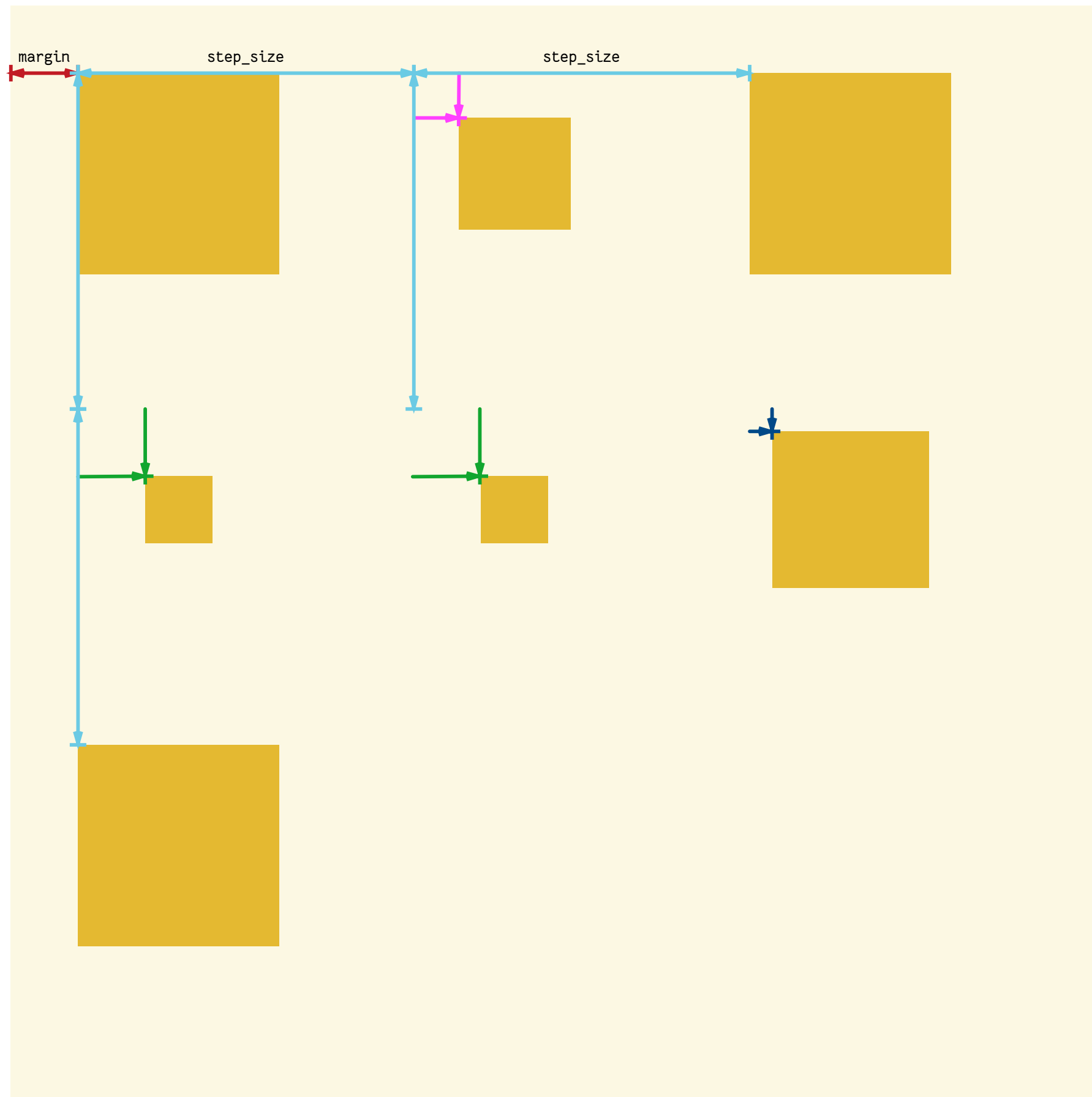
Ex. 5: Osynlig fyrkant runt fyrkanten. Mellanrum → 2x marginal



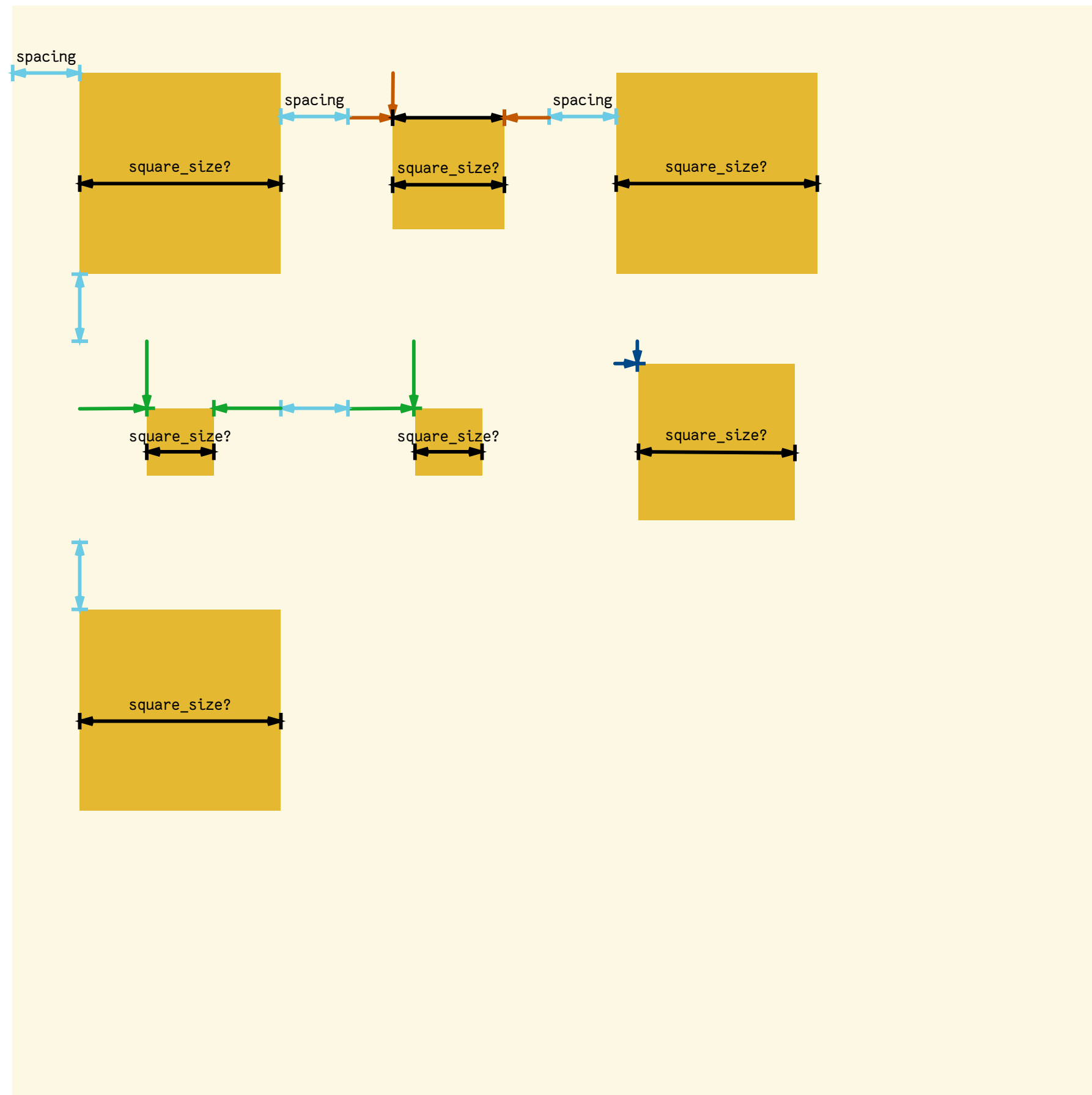
Ex. 6: Slumpmässig storlek



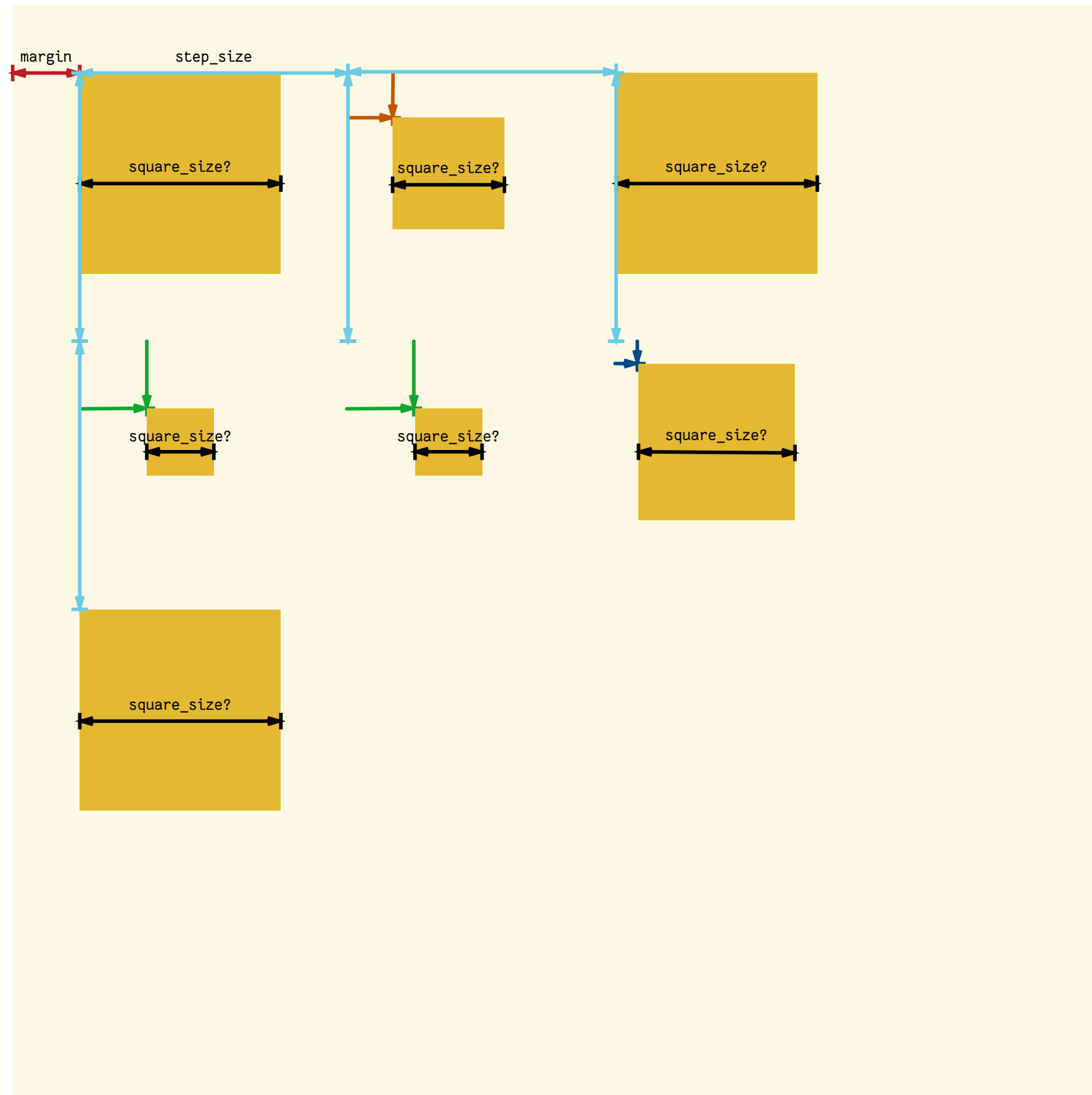
Ex. 7: Slumpmässig storlek



Ex. 8: Slumpmässig storlek



Ex. 9: Slumpmässig storlek



Ex. 10: Slumpmässig storlek

