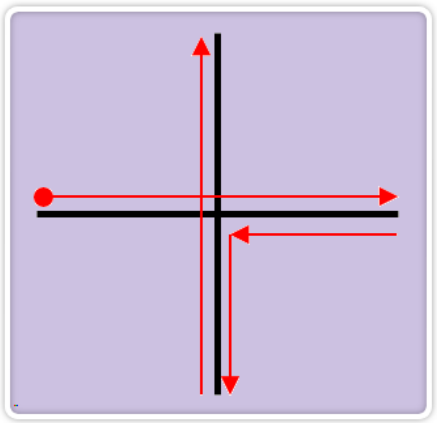
Урок 30. Поняття про складність алгоритмів

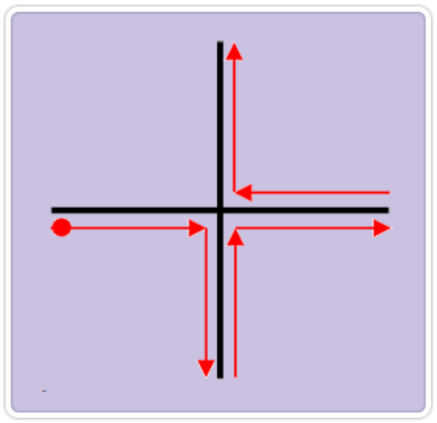
Вправа 2. Створи програму, що малює хрест за допомогою 7 команд модуля **turtle**.



Створи програму, у якій черепашка малює хрест, рухаючись зображеним маршрутом. Початок позначено кружком.

Готово

Вправа 3. Створи програму, що малює хрест за допомогою 5 команд модуля **turtle**, зокрема циклу.



Створи програму, у якій черепашка малює хрест, рухаючись зображеним маршрутом. Скористайся циклом з трьома ітераціями.

Готово

Складність запису алгоритму не є єдиною і найважливішою мірою складності алгоритму. Більш важлива часова складність — тривалість виконання алгоритму на комп'ютері.

Щоб виміряти час виконання програми, у Python є спеціальні команди.

- import time — підключення модуля вимірювання часу time
- time.time() — функція, що повертає поточний момент часу, перетворений на число

Загалом вимірювання часу роботи програми може виглядати так:

```
import time          #підключення модуля вимірювання часу
start = time.time()  #збереження моменту початку роботи програми у змінній start
...                  # тут розміщують власне програму
end = time.time()    #збереження моменту кінця роботи програми у змінній end
print(end - start)   #виведення часу роботи програми в секундах
```

Часова складність алгоритму – це тривалість його виконання на комп'ютері. На різних комп'ютерах часова складність алгоритму може бути різною.

Далі

Отже, складність запису алгоритму і його часова складність — різні поняття.

Програми, які містять більше команд, можуть виконуватися як довше, так і швидше програм, які містять менше команд.

Є й інші способи вимірювати складність алгоритмів. Наприклад, ємнісна складність — це обсяг пам'яті, який використовує програма під час свого виконання.

Часова і ємнісна складність — дві головні характеристики алгоритмів.

Часова і ємнісна складність взаємозалежні: алгоритми, що використовують менше пам'яті, часто (але не завжди) працюють довше.

Далі