COMP110: Principles of Computing

12: Further Computational Mathematics for Games

Worksheet E

- Assembly programming (TIS-100)
- Due **week 13** (after Christmas — see timetable)

Quiz E

- Probability
- Due **week 13** (after Christmas — see timetable)

Final worksheet submission

- Check MyFalmouth for the deadline
- **Download all five** of your worksheet forks as zips
- **Extract** them into five separate folders
- **Re-compress** the five folders into a **single zip file**
- **Upload** this zip file to LearningSpace

Recursion and induction

A formula for summation

$$\sum_{i=1}^{n} i = \frac{1}{2}n(n+1)$$

- $n = 1$: $1 = \frac{1}{2} \times 1 \times 2$
- $n = 2$: $1 + 2 = \frac{1}{2} \times 2 \times 3 = 3$
- $n = 3$: $1 + 2 + 3 = \frac{1}{2} \times 3 \times 4 = 6$
- ...

Proving the formula

- We can verify the formula for individual values of n
- How do we **prove** it for **all** n?
- We can use **proof by induction**

Proving the formula

Base case

- $n = 1$: $1 = \frac{1}{2} \times 1 \times 2$

Inductive assumption

- $\sum_{i=1}^{k-1} i = \frac{1}{2}(k-1)k$

Therefore

- $\sum_{i=1}^{k} i = \left(\sum_{i=1}^{k-1} i\right) + k$
- $= \frac{1}{2}(k-1)k + k$ (by inductive assumption)
- $= \frac{1}{2}k^2 - \frac{1}{2}k + k$
- $= \frac{1}{2}k^2 + \frac{1}{2}k$
- $= \frac{1}{2}k(k-1)$

So **if** the formula works for $n = k - 1$, **then** it works for $n = k$

Completing the proof

- We know:
  - The formula works for $n = 1$
  - If the formula works for $n = k - 1$, then it works for $n = k$
- The formula works for $n = 1$
- Therefore the formula works for $n = 1 + 1 = 2$
- Therefore the formula works for $n = 2 + 1 = 3$
- Therefore the formula works for $n = 3 + 1 = 4$
- ...
- Therefore the formula works for all positive integers n

Exercise

Prove

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

Thinking inductively

- I want to prove something for all n
- Given k, if I had already proved $n = k - 1$ then I could prove $n = k$
- I can also prove $n = 1$
- Therefore by induction I can prove the result for all n

Thinking recursively

- I want to solve a problem
- If I already had a function to solve smaller instances of the problem, I could use it to write my function
- I can solve the smallest possible problem
- Therefore I can write a recursive function

Exercise

- **Write** a pseudocode function to calculate the total size of all files in a directory and its subdirectories
- You may use the following functions in your pseudocode:
  - LISTDIR(directory): return a list of names of all files and folders in the given directory
  - GETSIZE(filename): return the size, in bytes, of the given file
  - ISDIR(name), ISFILE(name): determine whether the given name refers to a file or a directory

```
procedure CALCDIRSIZE(directory)
    ...                                        ▷ return total size in bytes
end procedure
```