Algorithms and Data Structures

Exercise Session 1

https://n.ethz.ch/~ahmala/and

Peer Grading

- Deadline is Monday 23:59
- 1 point every week
- One exercise chosen every week
- This week Exercise 1.2
- Official Solutions of Exercise Sheets will be uploaded before 11am
- Send your work to another group
- Receive some group’s work and grade/correct it
- Find mistakes, note that on paper/digital, use a distinctive color
- If everything is correct, just write that’s solved correctly
- Send it back to ahmala@ethz.ch

Corrections

- You receive your corrections in a week
- Uploaded to polybox.ethz.ch
- Located in /Shared/And

Important Series

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} \tag{1}$$

$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 + \cdots n^2 = \frac{n(n+1)(2n+1)}{6} \tag{2}$$

$$\sum_{k=0}^{n} q^k = \frac{q^{n+1} - 1}{q - 1} \tag{3}$$

$$\sum_{k=0}^{3} 3^k = 3^0 + 3^1 + 3^2 + 3^3 = \frac{3^4 - 1}{3 - 1} = 40 \tag{4}$$

Exercise 1.1 *Guess the formula* **(1 point)**.

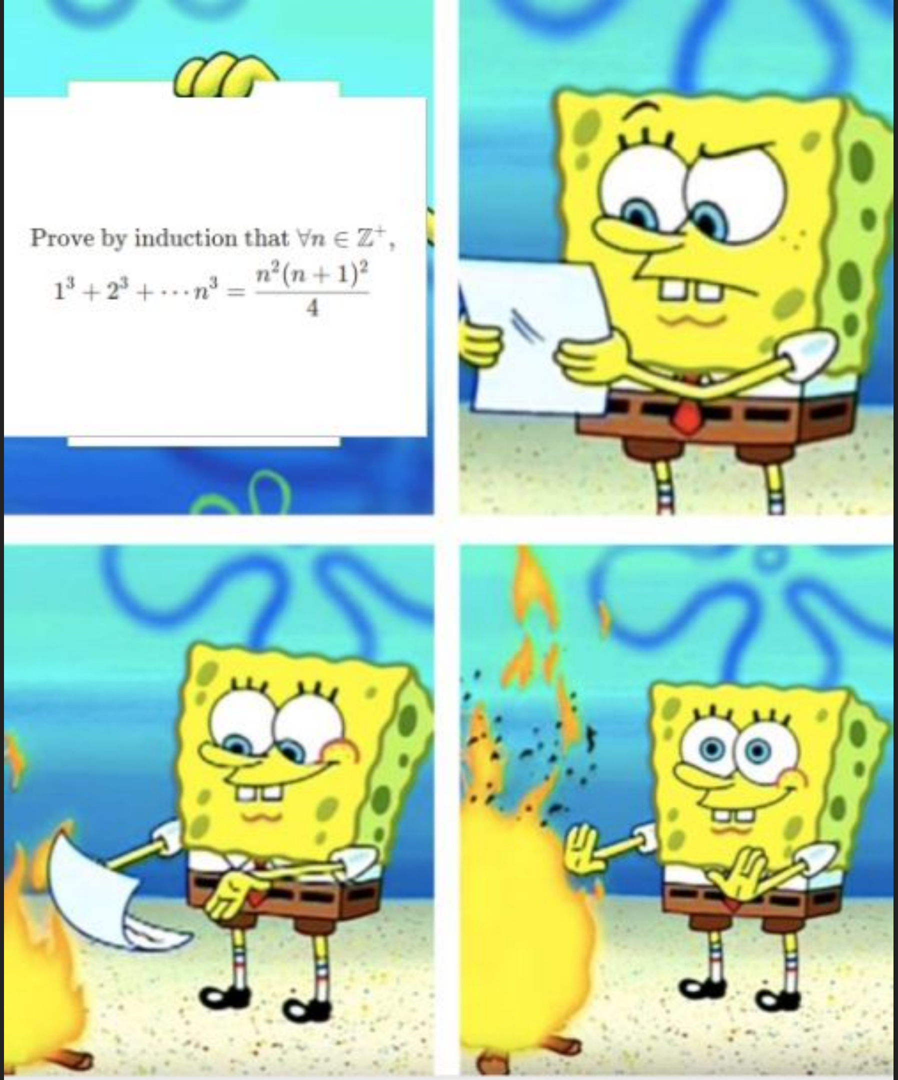
Consider the recursive formula defined by $a_1 = 2$ and $a_{n+1} = 3a_n - 2$ for $n > 1$. Find a simple closed formula for a_n and prove that a_n follows it using mathematical induction.

Hint: *Write out the first few terms. How fast does the sequence grow?*

Exercise 1.2 *Sum of Cubes* **(1 point)**.

Prove by mathematical induction that for every positive integer n,

$$1^3 + 2^3 + \cdots + n^3 = \frac{n^2(n+1)^2}{4}.$$

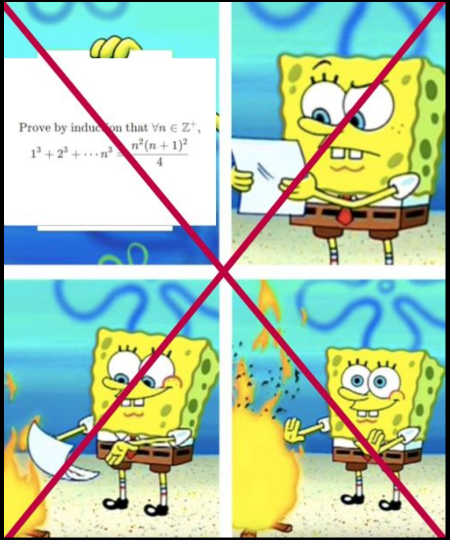


Prove by induction that $\forall n \in \mathbb{Z}^+$,
$1^3 + 2^3 + \cdots n^3 = \frac{n^2(n+1)^2}{4}$

Exercise 1.2 *Sum of Cubes* **(1 point)**.

Prove by mathematical induction that for every positive integer n,

$$1^3 + 2^3 + \cdots + n^3 = \frac{n^2(n+1)^2}{4}.$$



Exercise 1.3 *Sums of powers of integers.*

In this exercise, we fix an integer $k \in \mathbb{N}_0$.

(a) Show that, for all $n \in \mathbb{N}_0$, we have $\sum_{i=1}^{n} i^k \leq n^{k+1}$.

(b) Show that for all $n \in \mathbb{N}_0$, we have $\sum_{i=1}^{n} i^k \geq \frac{1}{2^{k+1}} \cdot n^{k+1}$.

Hint: *Consider the second half of the sum, i.e.,* $\sum_{i=\lceil \frac{n}{2} \rceil}^{n} i^k$*. How many terms are there in this sum? How small can they be?*

Together, these two inequalities show that $C_1 \cdot n^{k+1} \leq \sum_{i=1}^{n} i^k \leq C_2 \cdot n^{k+1}$, where $C_1 = \frac{1}{2^{k+1}}$ and $C_2 = 1$ are two constants independent of n. Hence, when n is large, $\sum_{i=1}^{n} i^k$ behaves "almost like n^{k+1}" up to a constant factor.

Example: When n is large $\sum_{i=1}^{n} i$ behaves almost like n^2 up to a constant factor.

Example: When n is large $\sum_{i=1}^{n} i^3$ behaves almost like n^4 up to a constant factor.

Exercise 1.4 *Asymptotic growth* **(1 point)**.

Recall the concept of asymptotic growth that we introduced in Exercise sheet 0: If $f, g : \mathbb{N} \to \mathbb{R}^+$ are two functions, then:

- We say that f *grows asymptotically slower than* g if $\lim_{m\to\infty} \frac{f(m)}{g(m)} = 0$. If this is the case, we also say that g *grows asymptotically faster than* f.

Prove or disprove each of the following statements.

(a) $f(m) = 10m^3 - m^2$ grows asymptotically slower than $g(m) = 100m^3$.

(b) $f(m) = 100 \cdot m^2 \log(m) + 10 \cdot m^3$ grows asymptotically slower than $g(m) = 5 \cdot m^3 \log(m)$.

Hint: $\log(m)$ *grows asymptotically slower than* m.

(c) $f(m) = \log(m)$ grows asymptotically slower than $g(m) = \log(m^4)$.

(d) $f(m) = 2^{(0.9m^2+m)}$ grows asymptotically slower than $g(m) = 2^{(m^2)}$.

(e) If f grows asymptotically slower than g, and g grows asymptotically slower than h, then f grows asymptotically slower than h.

Hint: *For any* $a, b : \mathbb{N} \to \mathbb{R}^+$, *if* $\lim_{m\to\infty} a(m) = A$ *and* $\lim_{m\to\infty} b(m) = B$, *then* $\lim_{m\to\infty} a(m)b(m) = AB$.

(f) If f grows asymptotically slower than g, and $h : \mathbb{N} \to \mathbb{N}$ grows asymptotically faster than 1, then f grows asymptotically slower than $g(h(m))$.

O-Notation

- Upper Bound
- useful in the analysis of algorithms

$$n \leq \mathcal{O}(n^2) \tag{1}$$

$$n \leq \mathcal{O}(n) \tag{2}$$

$$1 \leq \mathcal{O}(n) \tag{3}$$

$$log(n) \leq \mathcal{O}(\sqrt{n}) \tag{4}$$

$$100n^2 \leq \mathcal{O}(n^2) \tag{5}$$

$$\mathcal{O}(f) = \{g : \mathbb{N} \mapsto \mathbb{R}^+ | \, \exists C > 0 \, \forall n \in N \; g(n) \leq C \cdot f(n)\}$$

$$\mathcal{O}(n^2) = \{n, log(n), \sqrt{n}, log^2(n), 1, 100n^2, \cdots\}$$

Unit-Cost Random-Access Model

- Basic logical or arithmetic operations (+, *, =, if, call) are considered to be simple operations that take one time step.
- Loops and subroutines are complex operations composed of multiple time steps.
- All memory access takes exactly one time step.

O(n) operations

```
int ans = 0;
int i = 0;
for(i=0;i<n;i++){
	ans = ans + i;
}
```

O(n^2) operations

```
int ans = 0;
int i = 0;
int j = 0;
for(i=0;i<n;i++){
        for(j=0;j<n;j++)
                ans = ans + i + j;
}
```

O(n) operations

```
int ans = 0;
int i = 0;
int a[n];
for(i=0;i<n;i++){
        ans = ans + a[i];
}
```