Algorithms and Data Structures

Exercise Session 2

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

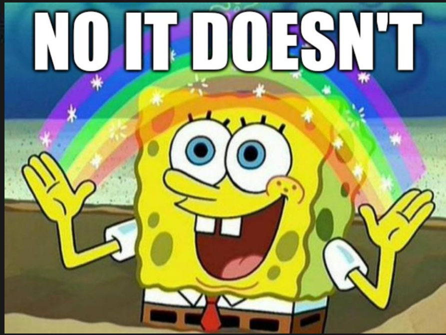
Common Mistakes

- Why does the following mean f(m) grow asymptotically slower than g(m)?

$$f(m) \in \mathcal{O}(g(m))$$

Common Mistakes

- Why does the following mean f(m) grow asymptotically slower than g(m)?



$$f(m) \in \mathcal{O}(g(m))$$

- Counter Example

$$f(m) = m, g(m) = m$$
$$f(m) \in \mathcal{O}(g(m))$$

Common Mistakes

Is the following correct?

$$\lim_{n\to\infty} A(n) = 0 \implies \lim_{n\to\infty} \Big(A(n) \cdot B(n)\Big) = 0$$

$$h : \mathbb{N} \mapsto \mathbb{N} \tag{1}$$

$$h \text{ grows asymptotically faster than } 1 \tag{2}$$

Do (1) and (2) mean h grows faster than or equal to $k(n) = n$?

$$h : \mathbb{N} \mapsto \mathbb{N} \tag{1}$$

$$h \text{ grows asymptotically faster than } 1 \tag{2}$$

Do (1) and (2) mean h grows faster than or equal to $k(n) = n$?

NO!

Consider $\lfloor \sqrt{n} \rfloor$

Is it possible to have more than one base case in induction?

Is it possible to have more than one base case in induction?

- P(n):= an n-cent postage can be made up using 3-cent and 7-cent stamps
- Prove P(n) for all n, where n is a natural number and n>=12 holds.

Source

Exercise 2.1 *Induction.*

(a) Prove via mathematical induction that for all integers $n \geq 5$,

$$2^n > n^2 .$$

(b) Let x be a real number. Prove via mathematical induction that for every positive integer n, we have

$$(1+x)^n = \sum_{i=0}^{n} \binom{n}{i} x^i \, ,$$

where

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \, .$$

We use a standard convention $0! = 1$, so $\binom{n}{0} = \binom{n}{n} = 1$ for every positive integer n.

Hint: *Y*ou can use the following fact without justification: for every $1 \le i \le n$,

$$\binom{n}{i} + \binom{n}{i-1} = \binom{n+1}{i} \, .$$

Theorem 1 (Theorem 1.1 from the script). *Let N be an infinite subset of $\mathbb{N}$ and $f : N \to \mathbb{R}^+$ and $g : N \to \mathbb{R}^+$.*

- *If* $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, *then* $f \leq O(g)$ *and* $g \not\leq O(f)$.
- *If* $\lim_{n\to\infty} \frac{f(n)}{g(n)} = C \in \mathbb{R}^+$, *then* $f \leq O(g)$ *and* $g \leq O(f)$.
- *If* $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$, *then* $f \not\leq O(g)$ *and* $g \leq O(f)$.

Theorem 2. *Let* $f, g, h : N \to \mathbb{R}^+$. *If* $f \leq O(h)$ *and* $g \leq O(h)$, *then*

1. *For every constant* $c > 0$, $c \cdot f \leq O(h)$.
2. $f + g \leq O(h)$.

Notice that for all real numbers $a, b > 1$, $\log_a n = \log_a b \cdot \log_b n$ (where $\log_a b$ is a positive constant). Hence $\log_a n \leq O(\log_b n)$. So you don't have to write bases of logarithms in asymptotic notation, that is, you can just write $O(\log n)$.

Exercise 2.2 *O-notation quiz.*

(a) For all the following functions the variable n ranges over $\mathbb{N}$. Prove or disprove the following statements. Justify your answer.

(1) $2n^5 + 10n^2 \leq O(\frac{1}{100}n^6)$

(2) $n^{10} + 2n^2 + 7 \leq O(100n^9)$

(3) $e^{1.2n} \leq O(e^n)$

(4)* $n^{\frac{2n+3}{n+1}} \leq O(n^2)$

$$n^{\frac{2n+3}{n+1}} \leq \mathcal{O}(n^2)$$

$$\lim_{n\to\infty} \frac{n^{\left(\frac{2n+3}{n+1}\right)}}{n^2}$$

$$= \lim_{n\to\infty} \frac{n^{\left(\frac{2n+2}{n+1}+\frac{1}{n+1}\right)}}{n^2}$$

$$= \lim_{n\to\infty} \frac{n^{\left(2+\frac{1}{n+1}\right)}}{n^2}$$

$$= \lim_{n\to\infty} \frac{n^2 n^{\left(\frac{1}{n+1}\right)}}{n^2}$$

$$= \lim_{n\to\infty} n^{\left(\frac{1}{n+1}\right)}$$

$$n^{\frac{2n+3}{n+1}} \leq \mathcal{O}(n^2)$$

$$\lim_{n\to\infty} \frac{n^{\left(\frac{2n+3}{n+1}\right)}}{n^2}$$

$$= \lim_{n\to\infty} \frac{n^{\left(\frac{2n+2}{n+1} + \frac{1}{n+1}\right)}}{n^2}$$

$$= \lim_{n\to\infty} \frac{n^{\left(2+\frac{1}{n+1}\right)}}{n^2}$$

$$= \lim_{n\to\infty} \frac{n^2 n^{\left(\frac{1}{n+1}\right)}}{n^2}$$

$$= \lim_{n\to\infty} n^{\left(\frac{1}{n+1}\right)}$$

$$\lim_{n\to\infty} n^{\left(\frac{1}{n+1}\right)}$$

$$\lim_{n\to\infty} n^{\left(\frac{1}{n+1}\right)}$$

$$= \lim_{n\to\infty} e^{\ln\left(n^{\left(\frac{1}{n+1}\right)}\right)}$$

$$= \lim_{n\to\infty} e^{\left(\frac{1}{n+1}\ln(n)\right)}$$

$$= \lim_{n\to\infty} e^{\frac{\ln(n)}{n+1}}$$

$$= e^{0}$$

$$= 1$$


(b) Find f and g as in Theorem 1 such that $f \leq O(g)$, but the limit $\lim_{n \to \infty} \frac{f(n)}{g(n)}$ does not exist. This proves that the first point of Theorem 1 provides a sufficient, but not a necessary condition for $f \leq O(g)$.
A: ...
B: ...
C: ...
D: ...

Exercise 2.3 *Asymptotic growth of* $\sum_{i=1}^{n} \frac{1}{i}$ **(1 point).**

The goal of this exercise is to show that the sum $\sum_{i=1}^{n} \frac{1}{i}$ behaves, up to constant factors, as $\log(n)$ when n is large. Formally, we will show $\sum_{i=1}^{n} \frac{1}{i} \leq O(\log n)$ and $\log n \leq O(\sum_{i=1}^{n} \frac{1}{i})$ as functions from $\mathbb{N}_{\geq 2}$ to $\mathbb{R}^+$.

For parts (a) to (c) we assume that $n = 2^k$ is a power of 2 for $k \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$. We will generalise the result to arbitrary $n \in \mathbb{N}$ in part (d). For $j \in \mathbb{N}$, define

$$S_j = \sum_{i=2^{j-1}+1}^{2^j} \frac{1}{i}.$$

(a) For any $j \in \mathbb{N}$, prove that $S_j \leq 1$.

Hint: *Find a common upper bound for all terms in the sum and count the number of terms.*

(b) For any $j \in \mathbb{N}$, prove that $S_j \geq \frac{1}{2}$.

(c) For any $k \in \mathbb{N}_0$, prove the following two inequalities

$$\sum_{i=1}^{2^k} \frac{1}{i} \leq k+1$$

and

$$\sum_{i=1}^{2^k} \frac{1}{i} \geq \frac{k+1}{2}.$$

Hint: *You can use that* $\sum_{i=1}^{2^k} \frac{1}{i} = 1 + \sum_{j=1}^{k} S_j$. *Use this, together with parts (a) and (b), to prove the required inequalities.*

(d)* For arbitrary $n \in \mathbb{N}$, prove that

$$\sum_{i=1}^{n} \frac{1}{i} \leq \log_2(n) + 2$$

and

$$\sum_{i=1}^{n} \frac{1}{i} \geq \frac{\log_2 n}{2}.$$

Hint: *Use the result from part (c) for* $k_1 = \lceil \log_2 n \rceil$ *and* $k_2 = \lfloor \log_2 n \rfloor$. *Here, for any* $x \in \mathbb{R}$, $\lceil x \rceil$ *is the smallest integer that is at least* x *and* $\lfloor x \rfloor$ *is the largest integer that is at most* x. *For example,* $\lceil 1.5 \rceil = 2$, $\lfloor 1.5 \rfloor = 1$ *and* $\lceil 3 \rceil = \lfloor 3 \rfloor = 3$. *In particular, for any* $x \in \mathbb{R}$, $x \leq \lceil x \rceil < x + 1$ *and* $x \geq \lfloor x \rfloor > x - 1$.

Exercise 2.4 *Asymptotic growth of* $\ln(n!)$.

Recall that the factorial of a positive integer n is defined as $n! = 1 \times 2 \times \cdots \times (n-1) \times n$. For the following functions n ranges over $\mathbb{N}_{\geq 2}$.

(a) Show that $\ln(n!) \leq O(n \ln n)$.

   Hint: *You can use the fact that* $n! \leq n^n$ *for* $n \in \mathbb{N}_{\geq 2}$ *without proof.*

(b) Show that $n \ln n \leq O(\ln(n!))$.

   Hint: *You can use the fact that* $\left(\frac{n}{2}\right)^{\frac{n}{2}} \leq n!$ *for* $n \in \mathbb{N}_{\geq 2}$ *without proof.*

Exercise 2.5 *Testing equations* **(2 points).**

Your friend sends you a piece of code that computes his favorite function $f : \mathbb{N} \to \mathbb{N}$. For $n \in \mathbb{N}$, we want to test if the equation $f(a) + f(b) + f(c) = f(d)$ can be satisfied using positive integers $1 \leq a, b, c, d \leq n$. Your friend completed Algorithms and Data Structures last year, and so you may assume that his code computes $f(k)$ in $O(1)$ for any $k \in \mathbb{N}$. You may also assume simple arithmetic operations on integers can be performed in $O(1)$. Finally, you may initialize an array of size k in time $O(k)$.

(a) Design a simple $O(n^4)$ algorithm that outputs "YES" if there exist integers $1 \leq a, b, c, d \leq n$ such that $f(a) + f(b) + f(c) = f(d)$ and "NO" otherwise.

(b) Assume that $f(k) \leq k^3$ for all $k \in \mathbb{N}$. Modify your previous algorithm so that it works in time $O(n^3)$ under this assumption. Motivate briefly why it still works.

Hint: *You could use a helper array of size* n^3 *to get rid of one of the loops in your previous algorithm. The helper array could save which values the function* f *can take.*

(c)* Assume that $f(k) \leq k^2$ for all $k \in \mathbb{N}$. Modify your previous algorithm so that it works in time $O(n^2)$ under this assumption. Motivate briefly why it still works.

Hint: *You could use a helper array again. Note that* $f(a) + f(b) + f(c) = f(d)$ *implies that* $f(a) + f(b) = f(d) - f(c)$.

Bonus Exercises Next Week

(a) Prove or disprove the following statements. Justify your answer.

(1) $\frac{1}{5}n^3 \geq \Omega(10n^2)$

(2) $n^2 + 3n = \Theta(n^2 \log(n))$

(3) $5n^4 + 3n^2 + n + 8 = \Theta(n^4)$

(4) $3^n \geq \Omega(2^n)$

(b) Prove the following statements.

Hint: *For these examples, computing the limits as in Theorem 1 is hard or the limits do not even exist. Try to prove the statements directly with inequalities as in the definition of the O-notation.*

(1) $(\sin(n) + 2)n = \Theta(n)$

Hint: *For any $x \in \mathbb{R}$ we have $-1 \leq \sin(x) \leq 1$.*

(2) $\sum_{i=1}^{n} \sum_{j=1}^{i} j = \Theta(n^3)$

Hint: *In order to show $n^3 \leq O(\sum_{i=1}^{n} \sum_{j=1}^{i} j)$, you can use exercise 1.3.*

(3) $\log(n^4 + n^3 + n^2) \leq O(\log(n^3 + n^2 + n))$

(4)* $\sum_{i=1}^{n} \sqrt{i} = \Theta(n\sqrt{n})$

Hint: *Recall again exercise 1.3 and try to do an analogous computation here.*

Exercise 3.3 *Counting function calls in loops* **(1 point)**.

For each of the following code snippets, compute the number of calls to f as a function of $n \in \mathbb{N}$. Provide **both** the exact number of calls and a maximally simplified asymptotic bound in Θ notation.

(a)

Algorithm 1

```
i ← 0
while i ≤ n do
    f()
    f()
    i ← i + 1
j ← 0
while j ≤ 2n do
    f()
    j ← j + 1
```

(b)

Algorithm 2

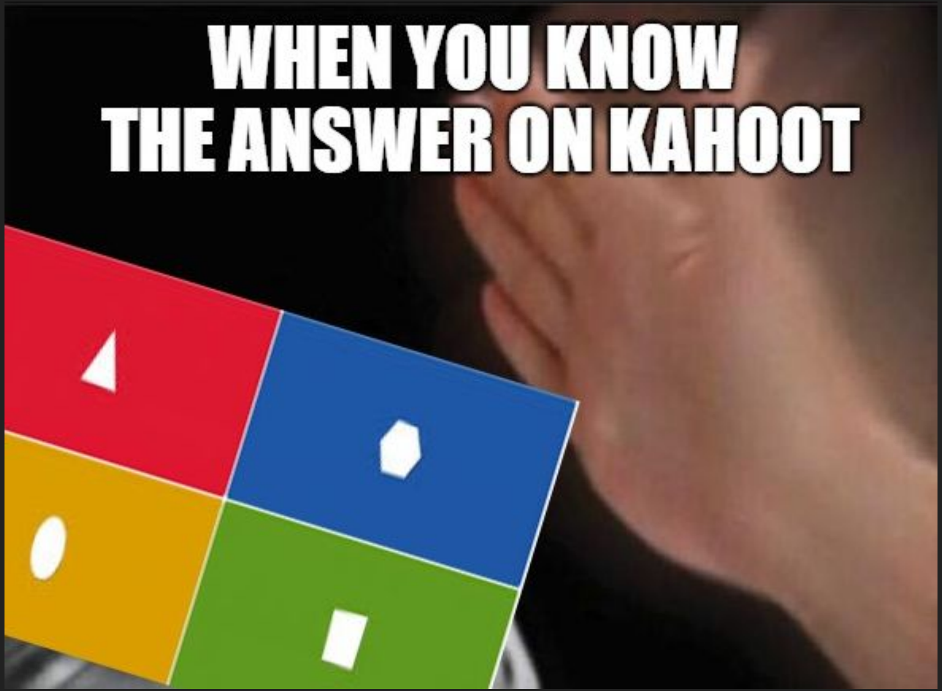
```
i ← 1
while i ≤ n do
    j ← 1
    while j ≤ i^3 do
        f()
        j ← j + 1
    i ← i + 1
```

Hint: *See Exercise 1.4.*

Maximum Subarray Sum Problem

https://cses.fi/problemset/task/1643/

Kahoot!



Peer Grading

Exercise 2.5.a and 2.5.b