

Porteføljeopgave 1

Mathias Balling Christiansen

October 21, 2024

Contents

1 Opgave	2
2 Opgave	2
3 Opgave	2
4 Opgave	2
5 Opgave	2
6 Opgave	2
7 Opgave	2
8 Opgave	3
9 Opgave	3
10 Opgave	3
11 Opgave	3
12 Opgave	3

1 Opgave

Jeg har implementeret algoritmen så ved lige tal trækkes 1 fra. Og ved ulige tal trækkes 2 fra for at ramme det næste ulige tal. Derved spares uoverflødige kald. Algoritmen stopper ved $n=1$.

2 Opgave

3 Opgave

Jeg har implementeret algoritmen så den stopper ved at dens længde er under 3 (basecase), da der skal bruges 3 tal til at sammenligne summen af 2 tal med den 3. Jeg tjekker bogstavets talværdi ved at trække ascii værdien af 0 fra. Hvis summen af 2 tal er lig med det 3. tal returneres 'true'. Hvis længden stadig er over 3 og summen ikke gav det 3. tal, kaldes funktionen med samme tekst, men hvor det første tal er fjernet.

4 Opgave

Her implementeres en algoritme der finder hvilke 3 tal i et array, der giver det tætteste tal på en potens af 2. For at optimere den har jeg et tjek der ser om vi rammer en præcis potens af 2. Hvis vi gør det, kan vi ikke komme tættere og derfor returneres. Tidskompleksiteten af algoritmen er $O(n^3)$, på grund af de 3 for-loops. For at få lavere tidskompleksitet algoritmen kunne alle summene af to tals gemmes i et hash map. Herefter kunne der itereres over alle værdier i dette, hvor det sidste tal summere og sammenlignes med potens af 2. Denne algoritme ville have 2 dobbel for-loops i sekvens, hvilket er $O(n^2)$

5 Opgave

6 Opgave

Her implementeres en algoritme der finder summer tal der er dividerbare med 3 i intervallet $[0, N]$. Hvis tallet er 0 returneres 0 (basecase). Hvis tallet er dividerbart med 3, returneres tallet og vi kalder algoritmen igen med tallet minus 3. Hvis tallet ikke er dividerbart med 3, returneres tallet minus (tallet modulus 3). Ved at trække tallet modulus 3 fra ved vi at vi rammer et tal der er dividerbart med 3 næste gang.

```
int logTo(int N) {
    if (N < 2)
        return 0;
    return 1 + logTo(N / 2);
}
```

7 Opgave

```
/*
Skriv en rekursiv metode med følgende signatur:
int logTo (int N)
Algoritmen returner totals-logaritmen af N, og det er en forudsætning, at N er
et naturligt tal og en potens af 2. Kaldt med N = 32 returneres 5, Og med N =
4096 returneres 12.
*/
#include <cassert>
#include <print>
```

```
int logTo(int N) {
    if (N < 2)
        return 0;
    return 1 + logTo(N / 2);
}

int main() {
    assert(logTo(4096) == 12);
    assert(logTo(50) == 5);
    assert(logTo(32) == 5);
    assert(logTo(3) == 1);
    assert(logTo(2) == 1);
    assert(logTo(1) == 0);

    std::println("All test passed!");
    return 0;
}
```

8 Opgave

9 Opgave

10 Opgave

11 Opgave

12 Opgave