

Opgave 1

Skriv en rekursiv algoritme, som har et naturligt tal som parameter og returnerer summen af de ulige tals kvadrater fra 1 til N.

Eksempel: kaldt med parameteren 8 returneres 84 ($1^2 + 3^2 + 5^2 + 7^2$).

Det er vigtigt at optimere algoritmen, så overflødige rekursive kald undgås.

Opgave 2

Hvad er Store-O tidskompleksiteten for nedenstående algoritme. Begrund dit svar.

```
public static int myMethod( int N )
{
    int x = 0; int y = 0;
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            for (int k = 0; k < N*Math.sqrt(N); k++)
                //square root; C++: #include <math.h>
            {
                x++;
            }
            j *= 2;
        }
        i += i;
    }
    for (int i = 0; i < N*N; i++)
        y++;
    return x+y;
}
```

Opgave 3

Skriv en *rekursiv* algoritme med følgende signatur:

```
bool additive(String s)
```

Parameteren indeholder en streng bestående af cifre, fx. "82842605".

Algoritmen returnerer true, hvis parameteren indeholder en substring af tre på hinanden efterfølgende tal, hvor det tredje ciffer er lig med summen af de to forrige.

I ovenstående eksempel returneres *true*, fordi indeks 5 (6) er summen af indeks 3 og 4 (4 plus 2).

Tip: ASCII-værdien af karakteren '7' er 55.

Opgave 4

Skriv en algoritme, der har et array af usorterede, entydige naturlige tal som input og find de tre tal i arrayet, hvis sum er tættest på en potens af 2. Det samme tal kan må bruges én gang.

Algoritmens returværdi skal være et heltalsarray, som først indeholder de tre tal og dernæst den tilhørende potens af to (fx 512).

Kaldt med arrayet {23,56,22,11,65,89,3,44,87,910,45,35,98}, returneres de tre tal 89, 3, 35 og potensen af 2: 128.

Hvad er Store-O tidskompleksiteten af din algoritme? Begrund dit svar og diskuter mulighederne for at optimere din løsning yderligere.

Opgave 5

Hvad er Store-O tidskompleksiteten af nedenstående metode? Begrund dit svar.

```
int myMethod(int N)
{
    int x = 0;
    for (int i = 1; i <= Math.sqrt(N); i++) //square root; C++: #include <math.h>
    {
        for (int j = 1; j <= N; j++)
        {
            for (int k = 1; k < N;)
            {
                x++;
                k = k * 2;
            }
        }
    }
    return x;
}
```

Opgave 6

Skriv en rekursiv algoritme med følgende signatur:

```
int sumDivisibleBy3(int N)
```

Algoritmen returnerer summen af heltal større end 0 og mindre end eller lig med N, som er dividérbare med 3.

Kaldt med $N = 12$ er den korrekte returværdi 30 (3+6+9+12). Kaldt med $N = 14$ er den korrekte returværdi også 30.

Din algoritme skal optimeres således, at overflødige rekursive kald undgås.

Opgave 7

6561 er et eksempel på et naturligt tal, som kan skrives som X^Y hvor X og Y er heltal, dvs. 9^4 . Et andet eksempel er 3125 (5^5).

Skriv en algoritme, som kan afgøre om et givent naturligt tal $Z < 100,000$ (algoritmens parameter) kan skrives som $Z=X^Y$ hvor X og Y er heltal, hvorom det gælder, at $X > 2$ og $Y > 2$.

Algoritmens returværdi skal designes således, at følgende information kan udledes fra den:

- Værdien af X (i tilfældet 6561 er $X=9$).
- Værdien af Y (i tilfældet 6561 er $Y=4$).
- For det givne Z kan der være flere løsningspar (X,Y) . For 3125 er der kun et løsningspar; men for 6561 er $X=3$ og $Y=8$ også en mulighed. Hvis der er mere end én løsning, skal den med den største X -værdi returneres.
- Hvis det givne Z ikke har noget løsningspar (X,Y) eller Z indeholder en ulovlig værdi, returneres en dummy eller default værdi.

Algoritmen skal optimeres under antagelsen af, at maksimumværdien for Z og minimumsværdierne for X og Y aldrig ændrer sig.

Opgave 8

Tabellen nedenfor er en hash tabel, hvor der anvendes quadratic probing til collision resolution.

Indeks Værdi

0	
1	
2	V
3	R
4	
5	
6	P
7	
8	E
9	
10	F

Indeks 0, 1, 4, 5, 7 og 9 er ledige. Vis hvordan tabellen ser ud, efter at elementerne Q, C and H, som hasher til henholdsvis indeks 7, 8 and 2 er indsat. Begrund dit svar.

Opgave 9

Hvad er Store-O tidskompleksiteten for nedenstående metode? Dit svar må godt være baseret på udførte tests. Hvis du vælger at give et svar uden tests, skal svaret begrundes.

```
public static long myMethod(int n)
{
    if (n <= 1)
        return 1;
    else
        return myMethod(n-1) + myMethod(n-2);
}
```

Opgave 10

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.

Opgave 11

Tabellen nedenfor repræsenterer de afgivne stemmer ved et valg.

{7,4,3,5,3,1,6,4,5,1,7,5}

I dette eksempel er der 7 kandidater (1-7), og der er afgivet 12 stemmer. Kandidat 6 fik 1 stemme, kandidaterne 1, 3, 4 og 7 fik hver 2 stemmer, kandidat 5 fik 3 stemmer, and kandidat 2 fik 0 stemmer.

Opgaven går ud på at skrive en algoritme, som kaldt med tabellen og eventuelt tabellens længde, kan afgøre om en kandidat fik mere end 50 % af stemmerne. I så fald returneres kandidatens nummer. Hvis ingen kandidat fik over 50 % af stemmerne, returneres -1.

I eksemplet opnåede ingen af kandidaterne flertal, og der returneres -1.

Hvad er din algoritmes tidskompleksitet?