

Uge 1 opgaver

Emil B Henriksen

5. september 2018

Resumé

A review of this weeks exercises. Algorithms and pseudo code.

Kapitel 1

Del 1

1.1 a

Vi skal bestemme hvad algorithmen udspytter med forskellige parameter
Ved at gennemgå algoritmen med de givende parametre kan vi bestemme om den returnerer True eller False.

```
1  exsist(A,n,x):  
2      lo = 0  
3      hi = n-1 #  
4      while hi >= lo :  
5          mid = floor((hi+lo)/2)  
6          if x > A[mid]:  
7              lo = mid+1  
8          else if x < A[mid]:  
9              hi = mid-1  
10         else :  
11             return true  
12     return False  
13  
14
```

Linje 2 indeholder funktionens navn, og de parameter den tager. A er en array, n er, muligvis, arrayens længde og x er en integer vi ønsker at tjekke om findes i den givende array. På linje 3 sættes variablen lo til 0, og på linje 4 bliver hi sat til lægnden af arrayen minus en. På 5 linje startes vores while loop, der køre så længe at konditionen er sand, at hi er højere eller lig lo.

Linje 6 Variablen mid sættes til $hi + lo$ divideret med 2, og så rundet ned, eg. $2.5 = 2$, $2.9 = 2$. Linje 7 bliver x tjekket om det er større end værdien som er i arrayen på index plads mid $A[mid]$. Hvis det er sandt, bliver lo sat til $mid + 1$ og loopet gentages med de nye værdier. Hvis x er mindre end $A[mid]$ køres den anden condition, hi bliver sat til $mid - 1$, og loopet gentages.

Hvis de to værdier skulle være lige store, vil det sidste forhold blive kørt,

som returnere True på linje 12. Hvis loopet aldrig finder det ønskede element, eller at der ikke søges i hele længden af arrayen, vil der returneres false.

```
exists(A, 8, 17)
returns true
```

1.2 b

```
exists(A, 8, 9)
returns false
```

1.3 c

```
exists(A, 4, 12)
returns false
```

1.4 d

mids værdier med parameterne (A,8,2)

loop	antal
0	3
1	1
2	0
3	return false

Kapitel 2

Del 2

Algoritmen er ment til at gennemkøre hele arrayen for at se om x er tilstede i arrayen parameteren n giver et startspunkt for algoritmen, men grundet algoritmens opbygning vil den også tjekke 'bag sig' for den værdi der checkes efter. En af grundene til at algoritmen fungerer er at den allerede er sorteret, en usorteret array ville give et udfald der ikke er forudsigeligt, da den er baseret på en sorteret array af positive integers \mathbb{Z}^+

Kapitel 3

Del 3

Hvis vi antager at A er en usorteret array, lad os antage en array B med følgende integers (stadigvæk positive integers) [20, 5, 1, 6, 9, 8, 17, 0]

(a) Jeg kan ikke lige se et tilfælde hvor True ville blive returneret, som i en falsk positiv.

(b) Når funktionen exists() kaldes med parameterne (B, 4, 20) vil algoritmen starte med at give "mid"værdien 2, x der har værdien 2 sammenlignes nu med B[2] hvilket har værdien 1. Det føre til at lo sættes til 2, og loopet vil køre igennem endnu engang siden $hi = 2$ og $lo = 2$, mid sættes til 2 igen, hvilket gør at første if statement køres igen. Variablen lo incrementeres med 1 og er nu 3, og derved afbrydes while løkken siden lo er højere en hi, og funktionen returnerer False.

Kapitel 4

Del 4

Hvis $n = 64$, vil loopet kun køre en gang, før at vi får en `indexerror`. Siden vores array kun er 8 lang, vil den ikke kunne finde noget på pladsen `A[31]`. Det er `undefined behavoir`, men lad os sige at det er en array hvor alt udefineret er 0