

Uge 1 opgaver

Jens Evald-Schelde, Jonathan Gabal Christiansen og Emil B. Henriksen

5. september 2018

Abstract

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

1 Del

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 exist(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
```

Linje 1 indeholder funktionens navn, og de parameter den tager. "A" er en array, "n" er arrayens længde og "x" er den integer vi leder efter i arrayen. På linje 2 sættes variablen "lo" til 0, og på linje 3 bliver "hi" sat til længden af arrayen minus en. På linje 4 startes vores while loop, der kører så længe at forholdet er sandt, hvilket er at "hi" er højere eller lig "lo". Hvis det ikke skulle være tilfældet returneres False

Linje 5 variablen "mid" sættes til "hi" + "lo" divideret med 2, og så rundet ned, f.eks. $\text{floor}(2.5) == 2$, $\text{floor}(2.9) == 2$. På linje 6 bliver "x" tjekket om den er større end værdien som er på arrayens indeksplads svarende til værdien i variablen "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 else if blokken, "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 11. Hvis loopet aldrig finder det ønskede element, eller hvis der ikke søges igennem hele længden, vil der returneres False.

a exists(A, 8, 17)
returns true

b exists(A, 8, 9)
returns false

c exists(A, 4, 12)
returns false

d midts værdier med parameterne (A,8,2)

loop	antal
0	3
1	1
2	0
3	return false

2 Del

Algoritmen er ment til at gennemkøre en array, for at se om "x"er tilstede i arrayen. Parameteren n giver et startspunkt for algoritmen, og er samtidigt længden på arrayen. Grundet algoritmens opbygning vil den også tjekke 'bag sig' hvis den værdi der tjekkes er større end det søgte. En af grundene til at algoritmen fungerer, er, at den allerede er sorteret fra mindste til højeste værdi. En usorteret array ville give et udfald der ikke er forudsigeligt eller efter hensigten, da den er baseret på en sorteret array af positive integers \mathbb{Z}^+ .

3 Del

Lad os antage at en array B med en længde på 8, der har følgende integers (stadigvæk kun positive integers) som elementer $B = [20, 5, 1, 6, 9, 8, 17, 0]$.

(a) Vi kan ikke lige se et tilfælde hvor True ville blive returneret, uden at integeren findes i arrayen. En falsk positiv burde altså ikke være mulig.

(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 derfor False, selvom vi ved at 20 er gemt i B[0].

4 Del

Hvis $n = 64$, vil loopet kun påbegynde den første omgang, før at vi får en index error. Siden vores array kun er 8 elementer lang, vil den ikke kunne finde noget på pladsen for A[31]. Der vil derfor være en uventet fejl som ikke bliver håndteret, og algoritmen kan ikke komme videre.