EKSAMEN

Algoritmer og Datastrukturer

Torsdag 30. maj 2024, 9:00—11:00

Institut for Datalogi, Naturvidenskabelige Fakultet, Aarhus Universitet

Antal sider i opgavesættet (incl. forsiden): 14

Tilladte medbragte hjælpemidler: **Ingen.**

Kun det separate svarark skal udfyldes og afleveres.

Denne opgaveformulering skal *ikke* afleveres.

Vejledning og pointgivning

Dette eksamenssæt består af en mængde multiple-choice-opgaver.

Svarene på opgaverne angives på det **separate svarark som afleveres**.

For hver opgave er angivet opgavens andel af det samlede eksamenssæt.

Hvert delspørgsmål har præcist et rigtigt svar.

For hvert delspørgsmål må du vælge **max ét svar** ved at afkrydse den tilsvarende rubrik.

Et delspørgsmål bedømmes som følgende:

- Hvis du sætter kryds ved det rigtige svar, får du 1 point.
- Hvis du ikke sætter nogen krydser, får du 0 point.
- Hvis du sætter kryds ved et forkert svar, får du $-\frac{1}{k-1}$ point, hvor k er antal svarmuligheder.

For en opgave med vægt $v\,\%$ og med n delspørgsmål, hvor du opnår samlet s point, beregnes din besvarelse af opgaven som:

$$\frac{s}{n} \cdot v\,\%$$

Bemærk at det er muligt at få negative point for en opgave.

Opgave 1 (Asymptotisk notation, 6 %)

I det følgende angiver $\log n$ 2-tals-logaritmen af n.

	Ja	Nej	
$2n + 5n^2$ er $O(n^3)$	A	B	1.1
$2n$ er $O(n)$	A	B	1.2
$3n \cdot \log n$ er $O(2^{\log n})$	A	B	1.3
2^n er $O(n^2)$	A	B	1.4
$n!$ er $O(2^n)$	A	B	1.5
$2^{3 \log n}$ er $O(n^2)$	A	B	1.6
$\sqrt{n}$ er $O(\sqrt{n} \cdot \log n)$	A	B	1.7
$n \cdot \log n$ er $O(\sqrt{n})$	A	B	1.8
$7 \cdot 2^7$ er $O(n^3)$	A	B	1.9
3^n er $\Omega(n \cdot \log n)$	A	B	1.10
$(\log n)^3$ er $\Omega((\log n)^2)$	A	B	1.11
$\sqrt{n} + n \cdot \log n$ er $\Theta(n \cdot \log n)$	A	B	1.12

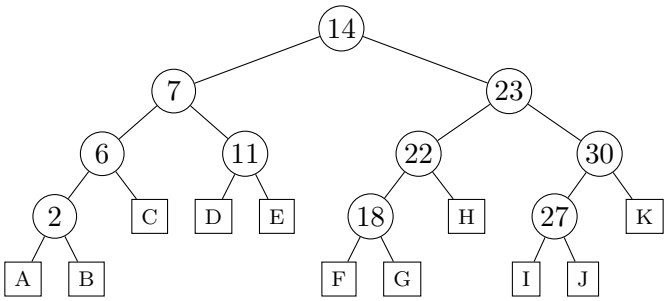
Opgave 2 (Analyse af løkker, 6 %)

```
Algoritme loop1(n)
  i = 1
  while i ≤ n
    i = 3 * i

Algoritme loop2(n)
  i = 1
  while i ≤ n^2
    i = i + i

Algoritme loop3(n)
  s = 0
  i = n
  while i > 0
    for j = 1 to i
      s = s + 1
    i = i − 1

Algoritme loop4(n)
  i = 1
  while i ≤ n
    j = 1
    k = 1
    while k ≤ i
      j = j + 1
      k = k + j
    i = 2 * i
```

Angiv for hver af ovenstående algoritmer udførselstiden som funktion af n i Θ-notation.

	$\Theta(n \log n)$	$\Theta((\log n)^2)$	$\Theta(\log n)$	$\Theta(n)$	$\Theta(n^3)$	$\Theta(n^2)$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n} \log n)$	
loop1	A	B	C	D	E	F	G	H	2.1
loop2	A	B	C	D	E	F	G	H	2.2
loop3	A	B	C	D	E	F	G	H	2.3
loop4	A	B	C	D	E	F	G	H	2.4

Opgave 3 (Indsættelser i søgetræer, 4 %)



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 4, 17, 20, 9 og 19 skal indsættes (det antages at før hver indsættelse indeholder træet kun ovenstående ti elementer).

	A	B	C	D	E	F	G	H	I	J	K	
INSERT(4)	A	B	C	D	E	F	G	H	I	J	K	3.1
INSERT(17)	A	B	C	D	E	F	G	H	I	J	K	3.2
INSERT(20)	A	B	C	D	E	F	G	H	I	J	K	3.3
INSERT(9)	A	B	C	D	E	F	G	H	I	J	K	3.4
INSERT(19)	A	B	C	D	E	F	G	H	I	J	K	3.5

Opgave 4 (Max-Heap-Insert, 4 %)

Angiv den binære max-heap efter indsættelse af elementerne 1, 3, 10, 5, 4, 12 og 9 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7	
1	3	10	5	4	12	9	A

1	2	3	4	5	6	7	
12	10	9	5	4	3	1	B

1	2	3	4	5	6	7	
12	5	10	3	4	1	9	C

1	2	3	4	5	6	7	
12	5	10	1	4	3	9	D

1	2	3	4	5	6	7	
10	5	12	3	4	1	9	E

Opgave 5 (Heap-Extract-Max, 4 %)

1	2	3	4	5	6	7	8	9	10	11	12	13
26	24	20	16	19	14	2	1	10	6	9	8	12

Hvad er resultatet af HEAP-EXTRACT-MAX på ovenstående max-heap ?

1	2	3	4	5	6	7	8	9	10	11	12	13	
24	19	20	16	9	14	2	1	10	6		8	12	A

1	2	3	4	5	6	7	8	9	10	11	12	
24	19	20	16	9	14	2	1	10	6	8	12	B

1	2	3	4	5	6	7	8	9	10	11	12	
24	19	20	16	9	14	2	1	10	6	12	8	C

1	2	3	4	5	6	7	8	9	10	11	12	
24	19	20	16	12	14	2	1	10	6	9	8	D

1	2	3	4	5	6	7	8	9	10	11	12	
24	20	16	19	14	12	1	10	6	9	8	2	E

Opgave 6 (Partition, 4 %)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
29	13	24	12	21	3	5	9	1	26	2	15	25	14	17

Angiv resultatet af at anvende PARTITION$(A, 4, 14)$ på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
29	13	24	12	3	5	9	1	2	14	21	15	25	26	17	A

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	2	3	5	9	12	13	14	15	17	21	24	25	26	29	B

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
29	13	24	1	2	3	5	9	12	14	15	21	25	26	17	C

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
29	13	24	12	3	5	9	1	2	14	21	26	15	25	17	D

Opgave 7 (Radix-sort, 4 %)

1340 0124 4440 1424 4421 1040

Betragt RADIX-SORT anvendt på ovenstående liste af tal ($d = 4$, $k = 5$). Angiv den delvist sortede liste efter at RADIX-SORT har sorteret tallene efter de *to* mindst betydende cifre.

1340	4440	1040	4421	0124	1424	A
0124	1040	1340	1424	4421	4440	B
4421	0124	1424	1040	1340	4440	C
4421	0124	1424	1340	4440	1040	D
0124	1040	1340	1424	4440	4421	E

Opgave 8 (Lineær probing, 4 %)

0	1	2	3	4	5	6	7	8	9	10
11	0							13	16	8

I ovenstående hashtabel af størrelse 11 er anvendt *lineær probing* med hashfunktionen $h(k) = 4k \bmod 11$.

Angiv positionerne de fem elementer 3, 4, 5, 7 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 8, 11, 13 og 16).

	0	1	2	3	4	5	6	7	8	9	10	
INSERT(3)	A	B	C	D	E	F	G	H	I	J	K	8.1
INSERT(4)	A	B	C	D	E	F	G	H	I	J	K	8.2
INSERT(5)	A	B	C	D	E	F	G	H	I	J	K	8.3
INSERT(7)	A	B	C	D	E	F	G	H	I	J	K	8.4
INSERT(9)	A	B	C	D	E	F	G	H	I	J	K	8.5

Opgave 9 (Kvadratisk probing, 4 %)

0	1	2	3	4	5	6	7	8	9	10
11			12		10			21	14	

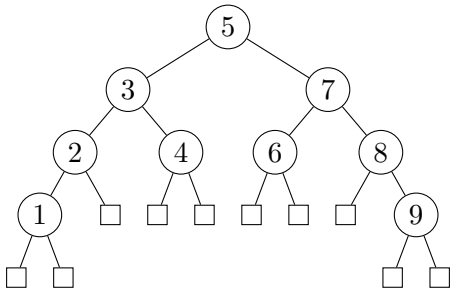
I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 3k \bmod 11$ og $h(k, i) = (h'(k) + 4i + 4i^2) \bmod 11$.

Angiv positionerne de fem elementer 0, 3, 6, 7 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 10, 11, 12, 14 og 21).

	0	1	2	3	4	5	6	7	8	9	10	
INSERT(0)	A	B	C	D	E	F	G	H	I	J	K	9.1
INSERT(3)	A	B	C	D	E	F	G	H	I	J	K	9.2
INSERT(6)	A	B	C	D	E	F	G	H	I	J	K	9.3
INSERT(7)	A	B	C	D	E	F	G	H	I	J	K	9.4
INSERT(9)	A	B	C	D	E	F	G	H	I	J	K	9.5

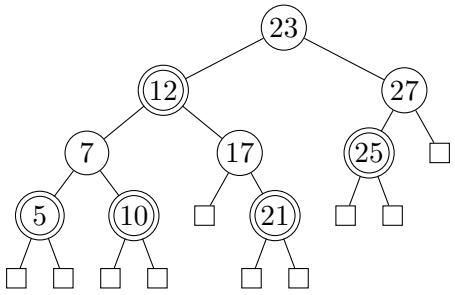
Opgave 10 (Rød-sorte træer, 4 %)

For hver af nedenstående delmængder, angiv om nedenstående binære træ er et lovligt rød-sort træ hvis netop disse knuder farves røde.

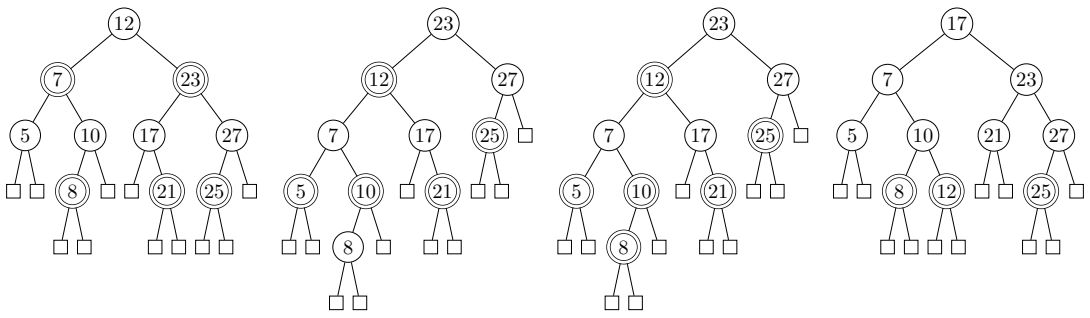


	Ja	Nej	
1, 3, 6, 8, 9	A	B	10.1
1, 9	A	B	10.2
1, 3, 7, 9	A	B	10.3
1, 2, 4, 7, 9	A	B	10.4
1, 5, 9	A	B	10.5

Opgave 11 (Indsættelse i rød-sorte træer, 4 %)



Angiv det resulterende rød-sorte træ når man indsætter 8 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).

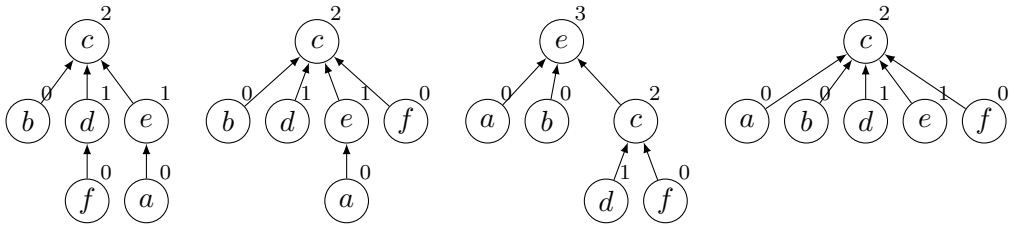


A B C D

Opgave 12 (Union-find, 4 %)

Angiv den resulterende union-find struktur efter nedenstående sekvens af operationer, når der anvendes union-by-rank og stikomprimering.

MAKESET(a)
MAKESET(b)
MAKESET(c)
MAKESET(d)
MAKESET(e)
MAKESET(f)
UNION(f, d)
UNION(b, c)
UNION(d, b)
UNION(a, e)
UNION(f, a)
FIND-SET(b)



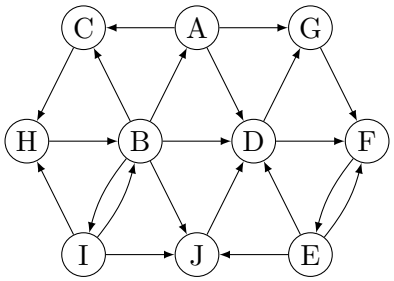
A B C D

Opgave 13 (Rekursionsligninger, 4 %)

Angiv løsningen for hver af nedenstående rekursionsligninger, hvor $T(n) = 1$ for $n \leq 1$.

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$	
$T(n) = 3 \cdot T(n/5) + n^3$	A	B	C	D	E	F	G	13.1
$T(n) = 8 \cdot T(n/2) + 1$	A	B	C	D	E	F	G	13.2
$T(n) = 9 \cdot T(n/3) + 1$	A	B	C	D	E	F	G	13.3
$T(n) = 2 \cdot T(n/4) + 3$	A	B	C	D	E	F	G	13.4
$T(n) = 9 \cdot T(n/3) + n^2$	A	B	C	D	E	F	G	13.5

Opgave 14 (BFS, 4 %)

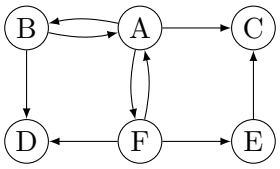


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

A D G C F H E B J I A C D G H F B E J I A C H B D F E J G I A C D G H F B E I J

A B C D

Opgave 15 (DFS, 4 %)



Betragt et dybde først gennemløb (DFS) af ovenstående graf, hvor DFS-gennemløbet starter i **knuden A**, hvor de udgående kanter til en knude besøges i alfabetisk rækkefølge. Angiv i hvilken rækkefølge knuderne får tildelt **finishing time**.

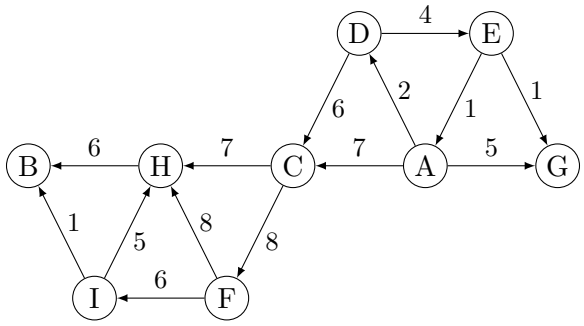
E D F C B A D B C E F A E F C D B A C D E F B A

A B C D 15.1

Angiv for hver af nedenstående kanter hvilken type kanten bliver i DFS gennemløbet.

	Tree edge	Back edge	Cross edge	Forward edge	
(A, C)	A	B	C	D	15.2
(B, D)	A	B	C	D	15.3
(B, A)	A	B	C	D	15.4
(F, D)	A	B	C	D	15.5

Opgave 16 (Dijkstras algoritme, 4 %)

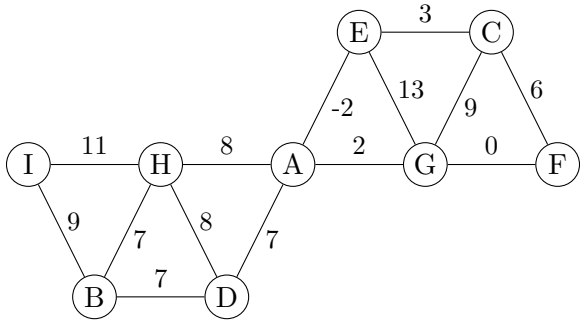


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

A D G E C H F B I A C D G F H E I B A D G E C H B F I A C F H B I D E G

A B C D

Opgave 17 (Prims algoritme, 4 %)



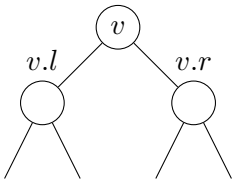
Antag Prims algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf, og algoritmen starter i **knuden A**. Angiv hvilken rækkefølge knuderne bliver inkluderet i det minimum udspændende træ (taget ud af prioritetskøen i Prims algoritme).

A E G F C D B H I A E C G F D B H I A E C G F D H B I A E C F G D B H I

A B C D

Opgave 18 (Udvidede søgetræer, 4 %)

Betragt en liste af n punkter $(x_1, y_1), \ldots, (x_n, y_n)$, hvor x_i og y_i er reelle tal og $x_1 < x_2 < \cdots < x_n$. Punkterne er gemt i et søgetræ, hvor hver knude v gemmer et punkt $(v.x, v.y)$, og knuderne er ordnet venstre-mod-højre efter stigende $v.x$. Derudover gemmes i en knude v værdien $v.maxy$ som er den maksimale y værdi af et punkt i v's undertræ.



Angiv hvorledes $v.maxy$ kan beregnes når den tilsvarende information er kendt ved de to børn $v.l$ og $v.r$ (det kan antages at disse begge eksisterer).

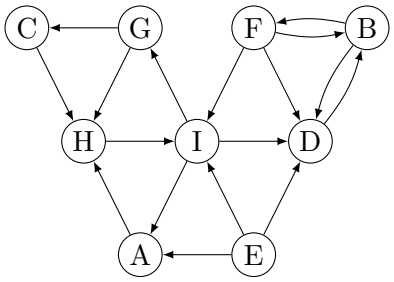
$v.maxy = \max(v.l.maxy, v.r.maxy)$ [A]

$v.maxy = \max(v.l.maxy, v.maxy, v.r.maxy)$ [B]

$v.maxy = \max(v.l.maxy, v.y, v.r.maxy)$ [C]

$v.maxy = \max(v.l.y, v.y, v.r.y)$ [D]

Opgave 19 (Stærke sammenhængskomponenter, 4 %)



Hvad er antallet af stærke sammenhængskomponenter i ovenstående graf?

1 2 3 4 5 6 7 8 9

[A] [B] [C] [D] [E] [F] [G] [H] [I]

Dynamisk programmering

Antag vi vil tælle antal måder at sætte lovlige paranteser på. F.eks. er nedenstående alle måder at sætte 3 par af lovlige paranteser.

()()() ()(()) (())() (()()) ((()))

Nedenstående er derimod eksempler på ulovlige måder at sætte 3 par af paranteser på.

)))((( )()()( )()(())(())(())(()

Vi lader P_n betegne antallet af lovlige måder at sætte n par af paranteser, dvs. $P_3 = 5$, og P_n kan bestemmes ved følgende rekursionsformel:

$$P_n = \begin{cases} 1 & \text{hvis } n = 0 \\ \sum_{i=0..n-1} P_i * P_{n-i-1} & \text{hvis } n > 0 \end{cases}$$

Opgave 20 (4 %)

Hvad er P_4?

5 6 7 8 9 10 11 12 12 14 15 16 17 18 19 20

A B C D E F G H I J K L M N O P

De følgende 2 opgaver består i at udfylde 2 blokke i følgende algoritmeskabelon.

Algoritme Paranteser(n)
Opret tabel $P[0\,..\,n]$
for ...
≪ **Opgave 21**: iterer over P ≫
≪ **Opgave 22**: beregn $P[i]$ ≫
Udskriv $P[n]$

Opgave 21 (4 %)

For hver af nedenstående stykker kode, angiv om det vil kunne føre til en korrekt løsning.

	Ja	Nej	
for $i = 0$ **to** $n - 1$	A	B	21.1
for $i = n - 1$ **to** 0 **step** -1	A	B	21.2
for $i = 0$ **to** n	A	B	21.3

Opgave 22 (4 %)

For hver af nedenstående stykker kode, angiv om det vil kunne føre til en korrekt løsning.

	Ja	Nej	

if $i = 0$ **then**
  $P[i] = 1$
else
  $P[i] = 0$
  for $j = i - 1$ **to** 0 **step** -1
  $P[i] = P[i] + P[j] * P[i - j - 1]$
	A	B	22.1

if $i = 0$ **then**
  $P[i] = 1$
else
  $P[i] = 0$
  for $j = 0$ **to** $i - 1$
  $P[i] = P[i] + P[j] * P[i - j - 1]$
	A	B	22.2

if $i = 0$ **then**
  $P[i] = 1$
else
  $P[i] = P[i - 1]$
  for $j = 1$ **to** $i - 1$
  $P[i] = P[i] + P[j] * P[i - j - 1]$
	A	B	22.3

Opgave 23 (Invarianter, 4 %)

Givet to ikke-negative heltal n og m, så beregner nedenstående algoritme $n + m$.

Algoritme Addition(n, m)
Inputbetingelse : Heltal $n \geq 0$ og $m \geq 0$
Outputkrav : $r = n_0 + m_0$
Metode : $r \leftarrow n$
$\{I\}$ **while** $m > 0$ **do**
$r \leftarrow r + 1$
$m \leftarrow m - 1$

For hvert af følgende udsagn, angiv om det er en invariant I for algoritmen Addition, hvor n_0 og m_0 angiver værdierne for henholdsvis n og m i starten.

	Ja	Nej	
$r = n + m$	A	B	23.1
$r = n_0 + m_0$	A	B	23.2
$r + m = n_0 + m_0$	A	B	23.3
$r + m_0 = n_0 + m_0$	A	B	23.4
$0 \leq m \leq m_0$	A	B	23.5

Opgave 24 (Amortiseret analyse, 4 %)

En decimal tæller består af n cifre $x_0, x_1, \ldots, x_{n-1}$, hvor $0 \leq x_i \leq 9$ for alle $i = 0, \ldots, n-1$. Værdien tælleren repræsenterer er $\sum_{i=0}^{n-1} x_i \cdot 10^i$. Nedenstående algoritme Inc øger værdien af tælleren med en, bortset fra hvis tælleren før Inc udføres repræsenterer værdien $10^n - 1$, så vil værdien efter Inc er udført være 0.

Algoritme Inc$(x_0, \ldots, x_{n-1})$
$i \leftarrow 0$
while $i < n - 1 \ \wedge \ x_i = 9$ **do**
$x_i \leftarrow 0$
$i \leftarrow i + 1$
$x_i \leftarrow (x_i + 1) \bmod 10$

Angiv for hver af nedenstående funktioner om det er en potentialefunktion, hvormed man kan argumentere for at Inc tager amortiseret $O(1)$.

	Ja	Nej	
$\lvert\{i \mid 0 \leq i < n \wedge x_i = 0\}\rvert$	A	B	24.1
$\lvert\{i \mid 0 \leq i < n \wedge x_i \neq 0\}\rvert$	A	B	24.2
$\lvert\{i \mid 0 \leq i < n \wedge x_i = 9\}\rvert$	A	B	24.3
$\max\{i \mid 0 \leq i \leq n \wedge x_0 \neq 0 \wedge x_1 \neq 0 \wedge \cdots \wedge x_{i-1} \neq 0\}$	A	B	24.4
$\max\{i \mid 0 \leq i \leq n \wedge x_0 = 9 \wedge x_1 = 9 \wedge \cdots \wedge x_{i-1} = 9\}$	A	B	24.5