EKSAMEN

Algoritmer og Datastrukturer

Maj 21, 2025

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 notationer, 6 %)

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

	Ja	Nej	
$\log n + n \cdot \log n$ er $O((\log n)^3)$	A	B	1.1
$n \cdot \log n$ er $O(n^{2/3})$	A	B	1.2
3^n er $O(2^n)$	A	B	1.3
$3n \cdot \log n$ er $O(2^n)$	A	B	1.4
$3\sum_{i=1}^{n} i$ er $O(n^2)$	A	B	1.5
$7(\log n)^3$ er $O(n^3)$	A	B	1.6
n er $O(\sqrt{n})$	A	B	1.7
$n \cdot (\log n)/6$ er $O(2^n)$	A	B	1.8
$n^{\log n}$ er $O((\log n)^n)$	A	B	1.9
$4^{\log n}$ er $\Omega(n \log n)$	A	B	1.10
$\log n$ er $\Theta(n^{1/3})$	A	B	1.11
$n \cdot \log n$ er $\Theta(n)$	A	B	1.12

Opgave 2 (Analyse af løkker, 6 %)

```
Algoritme loop1(n)
  i = 1
  while i ≤ n * n
      i = 2 * i
```

```
Algoritme loop2(n)
  s = 1
  for i = 1 to n
      for j = i to i + n
          for k = i to j
              s = s + 1
```

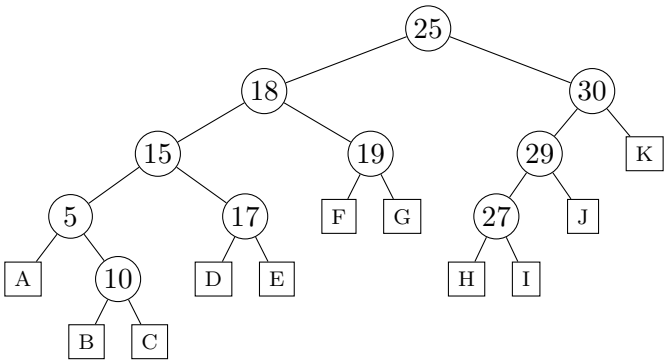
```
Algoritme loop3(n)
  i = 1
  j = n
  while i ≤ j
      i = 2 * i
      j = j + 1
```

```
Algoritme loop4(n)
  s = 1
  j = 1
  for i = 1 to n
      while j ≤ s
          j = j + 1
      s = 2 * s
```

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

	$\Theta((\log n)^2)$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(2^n)$	$\Theta(\sqrt{n})$	$\Theta(\log n)$	$\Theta(n^2)$	$\Theta(n^3)$	
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 (Indsættelse i søgetræer, 4 %)



Angiv i hvilke blade A–K i ovenstående ubalancerede binære søgetræ elementerne 23, 14, 7, 12 og 28 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(23)	A	B	C	D	E	F	G	H	I	J	K	3.1
INSERT(14)	A	B	C	D	E	F	G	H	I	J	K	3.2
INSERT(7)	A	B	C	D	E	F	G	H	I	J	K	3.3
INSERT(12)	A	B	C	D	E	F	G	H	I	J	K	3.4
INSERT(28)	A	B	C	D	E	F	G	H	I	J	K	3.5

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

1	2	3	4	5	6	7	8	9
7	1	6	5	8	4	2	9	3

Hvad er resultatet af BUILD-MAX-HEAP på ovenstående array ?

1	2	3	4	5	6	7	8	9	
9	8	7	6	5	4	3	2	1	A
7	8	6	9	1	4	2	5	3	B
1	2	3	4	5	6	7	8	9	C
9	8	6	7	5	4	2	1	3	D
9	8	6	5	7	4	2	1	3	E

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

1	2	3	4	5	6	7	8	9	10	11	12	13
24	22	16	10	11	15	2	9	4	6	3	13	12

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

1	2	3	4	5	6	7	8	9	10	11	12	
22	12	16	10	11	15	2	9	4	6	3	13	A

1	2	3	4	5	6	7	8	9	10	11	12	
22	11	16	10	6	15	2	9	4	3	13	12	B

1	2	3	4	5	6	7	8	9	10	11	12	
22	16	12	11	15	10	9	4	6	3	13	2	C

1	2	3	4	5	6	7	8	9	10	11	12	
22	11	16	10	6	15	2	9	4	12	3	13	D

1	2	3	4	5	6	7	8	9	10	11	12	13	
22	11	16	10	6	15	2	9	4		3	13	12	E

Opgave 6 (Partition, 4 %)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
27	6	4	25	26	28	1	3	7	29	24	19	2	16	10

Angiv resultatet af at anvende PARTITION$(A, 4, 13)$ på ovenstående array A.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
27	6	4	1	2	25	26	28	3	7	29	24	19	16	10	A

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
27	6	4	1	2	28	25	3	7	29	24	19	26	16	10	B

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	2	3	4	6	7	10	16	19	24	25	26	27	28	29	C

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
27	6	4	1	2	3	7	19	24	25	26	28	29	16	10	D

Opgave 7 (Lineær probing, 4 %)

0	1	2	3	4	5	6	7	8	9	10
					8		9	15	10	21

I ovenstående hashtabel af størrelse 11 er anvendt *lineær probing* med hashfunktionen $h(k) = 2k \bmod 11$.

Angiv positionerne de fem elementer 2, 4, 5, 6 og 7 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 8, 9, 10, 15 og 21).

	0	1	2	3	4	5	6	7	8	9	10	
INSERT(2)	A	B	C	D	E	F	G	H	I	J	K	7.1
INSERT(4)	A	B	C	D	E	F	G	H	I	J	K	7.2
INSERT(5)	A	B	C	D	E	F	G	H	I	J	K	7.3
INSERT(6)	A	B	C	D	E	F	G	H	I	J	K	7.4
INSERT(7)	A	B	C	D	E	F	G	H	I	J	K	7.5

Opgave 8 (Kvadratisk probing, 4 %)

0	1	2	3	4	5	6	7	8	9	10
0				3		22	19			13

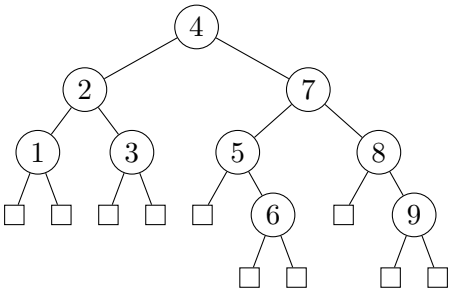
I ovenstående hashtabel af størrelse 11 er anvendt *kvadratisk probing* med hashfunktionerne $h'(k) = 5k \bmod 11$ og $h(k, i) = (h'(k) + 5i + i^2) \bmod 11$.

Angiv positionerne de fem elementer 1, 2, 4, 8 og 9 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 0, 3, 13, 19 og 22).

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

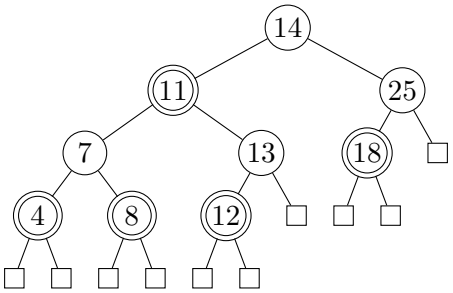
Opgave 9 (Rød-sorte træer, 4 %)

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

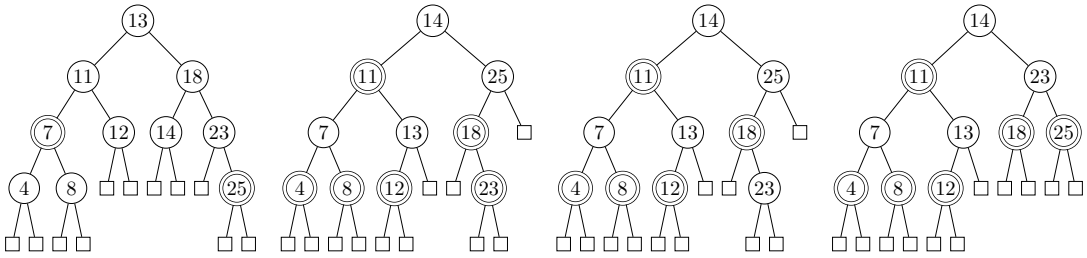


	Ja	Nej	
1, 3, 5, 6, 8, 9	A	B	9.1
2, 6, 7, 9	A	B	9.2
1, 3, 6, 7, 9	A	B	9.3
4, 6, 9	A	B	9.4
6, 9	A	B	9.5

Opgave 10 (Indsættelse i rød-sorte træer, 4 %)



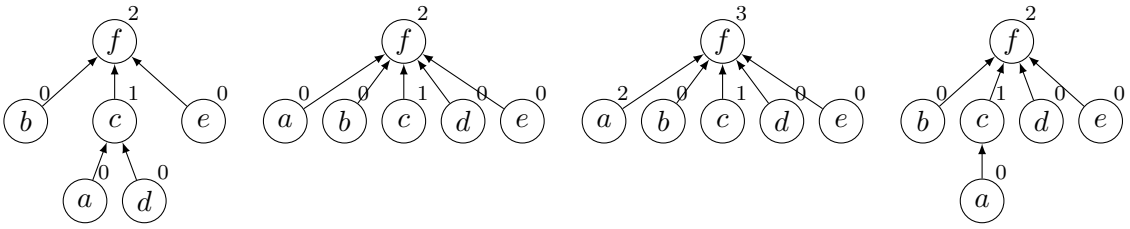
Angiv det resulterende rød-sorte træ når man indsætter 23 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).



Opgave 11 (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(d, c)
UNION(d, a)
UNION(e, f)
UNION(a, e)
UNION(b, d)
FIND-SET(b)



A B C D

Opgave 12 (Rekursionsligninger, 4 %)

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

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$	
$T(n) = 4 \cdot T(n/2) + 1$	A	B	C	D	E	F	G	12.1
$T(n) = 3 \cdot T(n/9) + 2$	A	B	C	D	E	F	G	12.2
$T(n) = 2 \cdot T(n/4) + n^3$	A	B	C	D	E	F	G	12.3
$T(n) = T(n-1) + n^2$	A	B	C	D	E	F	G	12.4
$T(n) = T(n-1) + \log n$	A	B	C	D	E	F	G	12.5

Opgave 13 (Heltalsmultiplikation, 4 %)

I analysen af Karatsubas del-og-kombiner algoritme for multiplikation af to lange heltal, hver med n bits, indgår hvilken af følgende rekursionsligninger?

$T(n) = 2 \cdot T(n/3) + n$ A

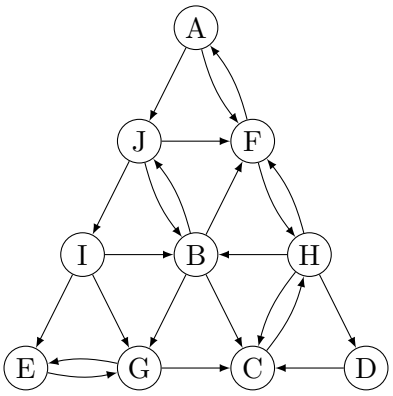
$T(n) = 3 \cdot T(n/2) + n$ B

$T(n) = 2 \cdot T(n/3) + n^2$ C

$T(n) = 3 \cdot T(n/2) + n^2$ D

$T(n) = 7 \cdot T(n/2) + n^2$ E

Opgave 14 (BFS, 4 %)

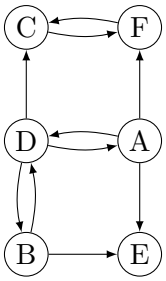


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

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

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 **discovery time**.

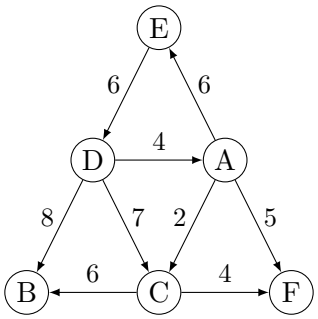
A D E F B C A E F C D B A D C F B E A D B E C F

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, E)	A	B	C	D	15.2
(B, D)	A	B	C	D	15.3
(F, C)	A	B	C	D	15.4
(A, D)	A	B	C	D	15.5

Opgave 16 (Dijkstras algoritme, 4 %)

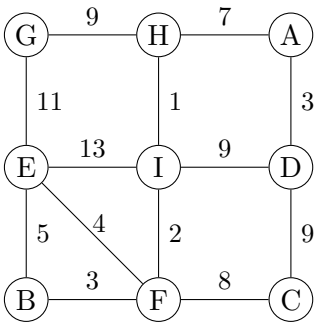


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstras algoritme.

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

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

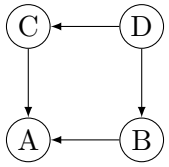
A B C D

Opgave 18 (Grafegenskaber, 4 %)

Antag vi har en orienteret graf $G = (V, E)$ og en startknude $s \in V$, hvor et BFS gennemløb af G startende i s kan nå alle knuder i V, og et BFS gennemløb af $\overline{G} = (V, \overline{E})$ startende i s kan nå alle knuder i V, hvor $\overline{G}$ er grafen G hvor alle kanter er modsatrettede. Hvad ved vi om grafen?

	Ja	Nej	
G er en acyklisk graf	A	B	18.1
Gennemløbet af $\overline{G}$ besøger knuderne i G i topologisk orden	A	B	18.2
G er stærkt sammenhængende	A	B	18.3
G har mindst to stærke sammenhængskomponenter	A	B	18.4

Opgave 19 (Topologisk sortering, 4 %)



Angiv for hver af nedenstående ordninger af knuderne i ovenstående graf om det er en lovlig topologisk sortering.

	Ja	Nej	
D A C B	A	B	19.1
D B C A	A	B	19.2
D C A B	A	B	19.3
D C B A	A	B	19.4
B C D A	A	B	19.5

Opgave 20 (Invarianter, 4 %)

Givet to heltal x og y, så beregner nedenstående algoritme $x - y$.

Algoritme Minus(x, y)
Inputbetingelse : Heltal x og y
Outputkrav : $r = x - y$
Metode : $r \leftarrow 0$
$\{I\}$ **while** $x \neq y$ **do**
if $x > y$ **then**
$y \leftarrow y + 1$
$r \leftarrow r + 1$
else
$x \leftarrow x + 1$
$r \leftarrow r - 1$

For hvert af følgende udsagn, angiv om det er en invariant I for algoritmen Minus, hvor x_0 og y_0 angiver værdierne for henholdsvis x og y i starten.

	Ja	Nej	
$x \neq y$	A	B	20.1
$x + y \geq x_0 + y_0$	A	B	20.2
$r = x - y$	A	B	20.3
$x - y = r + x_0 - y_0$	A	B	20.4
$x - y + r = x_0 - y_0$	A	B	20.5
$\min\{x_0, y_0\} \leq \min\{x, y\}$	A	B	20.6

Dynamisk programmering

Antag vi har to strenge $s = s_1 \cdots s_n$ og $t = t_1 \cdots t_m$, og vi ønsker at beregne den mindste *vægtede redigeringsafstand* mellem s og t, som er defineret som følgende. Startende med s, kan vi slette et tegn i s eller indsætte et nyt tegn imellem to eksisterende tegn i s (muligvis som et nyt første eller sidste tegn). Hver slettelse og indsættelse koster 6. Desuden kan vi udskifte et symbol a i s med et andet symbol b, hvor prisen for udskiftningen er $|\text{rank}(a) - \text{rank}(b)|$, hvor $\text{rank}(a)$ angiver position af a i alfabetet. Den vægtede redigeringsafstand er den mindst mulige pris for at komme fra s til t ved en sekvens af redigeringer. F.eks. er den vægtede redigeringsafstand mellem $s =$ `ABANANAMAN` og $t =$ `BANANARAMA` 18, da

pris	6						5		1	6
s	A	B	A	N	A	N	A	M	A	N	-
t	-	B	A	N	A	N	A	R	A	M	A

Lad $E_{i,j}$ betegne den vægtede redigeringsafstand mellem $s_{i+1} \cdots s_n$ og $t_{j+1} \cdots s_m$ (dvs. mellem s og t, hvor vi har fjernet henholdsvis de første i og j tegn), hvor $0 \leq i \leq n$ og $0 \leq j \leq m$, som kan beregnes ved følgende rekursionsformel:

$$E_{i,j} = \begin{cases} 6 \cdot (n + m - i - j) & \text{hvis } i = n \ \vee \ j = m \\ \min(6 + E_{i+1,j},\ 6 + E_{i,j+1},\ |\text{rank}(s_{i+1}) - \text{rank}(t_{j+1})| + E_{i+1,j+1}) & \text{ellers.} \end{cases}$$

De følgende 3 opgaver består i at udfylde 3 blokke i følgende algoritmeskabelon.

Algoritme Redigeringsafstand(s, t)
Opret tabel $E[0\,..\,n][0\,..\,m]$

```
for ...
        ≪ Opgave 21: iterer over E ≫
    ≪ Opgave 22: beregn E[i][j] ≫
≪ Opgave 23: Udskriv uændrede eller udskiftede tegn fra s til t ≫
```

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 **do** **for** $j = 0$ **to** m **do**	A	B	21.1
for $i = n$ **to** 0 **step** -1 **do** **for** $j = m$ **to** 0 **step** -1 **do**	A	B	21.2
for $j = m$ **to** 0 **step** -1 **do** **for** $i = n$ **to** 0 **step** -1 **do**	A	B	21.3

Opgave 22 (4 %)

For hver af nedenstående stykker kode, angiv om det vil kunne føre til en korrekt løsning, hvor $\delta(a, b) = |\text{rank}(a) - \text{rank}(b)|$.

Ja Nej

$E[i][j] = 6 * (m + n - i - j)$
if $i < n$ **and** $j < m$ **then**
  $E[i][j] = \min(6 + E[i+1][j],\ 6 + E[i][j+1],\ \delta(s_{i+1}, t_{j+1}) + E[i+1][j+1])$

A B 22.1

---

if $i = n$ **or** $j = m$ **then**
  $E[i][j] = 6 * (m + n - i - j)$
else
  $E[i][j] = \min(6 + E[i+1][j],\ 6 + E[i][j+1],\ \delta(s_{i+1}, t_{j+1}) + E[i+1][j+1])$

A B 22.2

$E[i][j] = 6 * (m + n - i - j)$
if $i < n$ **and** $j < m$ **then**
  $E[i][j] = \min(E[i][j],\ \delta(s_{i+1}, t_{j+1}) + E[i+1][j+1])$

A B 22.3

Opgave 23 (4 %)

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

Ja Nej

$i = 0$
$j = 0$
while $i < n$ **or** $j < m$ **do**
  if $i = n$ **or** ($i < n$ **and** $j < m$ **and** $E[i][j] = 6 + E[i][j+1]$) **then**
    $j = j + 1$
  else if $j = n$ **or** ($i < n$ **and** $j < m$ **and** $E[i][j] = 6 + E[i+1][j]$) **then**
    $i = i + 1$
  else
    $i = i + 1$
    $j = j + 1$
    output $s[i]$ bliver til $t[j]$

A B 23.1

$i = 0$
$j = 0$
while $i < n$ **and** $j < m$ **do**
  if $E[i][j] = \delta(s_{i+1}, t_{j+1}) + E[i+1][j+1]$ **then**
    $i = i + 1$
    $j = j + 1$
    output $s[i]$ bliver til $t[j]$
  else if $E[i][j+1] \leq E[i+1][j]$ **then**
    $j = j + 1$
  else
    $i = i + 1$

A B 23.2

Opgave 24 (Amortiseret analyse, 4 %)

Antag vi ønsker at vedligeholde en stigende liste af forskellige heltal under operationerne: InsertLeft(e) indsætter e først i listen efter at have slettet alle tal i listen der er mindre end eller lig med e; InsertRight(e) indsætter e sidst i listen efter at have slettet alle tal i listen der er større end eller lig med e; PopMin() sletter og returnerer det mindste tal i listen; og PopMax() sletter og returnerer det største tal i listen. F.eks. InsertRight(11) på listen $1, 3, 4, 7, 9, 10, 13, 14$ giver listen $1, 3, 4, 7, 9, 10, 11$.

Antag at listen gemmes i et *cyklist array*, f.eks.

1			*tail*				*head*			*N*
9	10	11					1	3	4	7

Lad n betegne det aktuelle antal elementer i listen og N størrelsen af det cykliske array. Når n bliver lig med N, så kopieres listen over i et nyt array af størrelse $2N$ i $O(N)$ tid. Angiv for hver af nedenstående funktioner om det er en potentialefunktion, hvormed man kan argumentere for at alle fire operationer tager amortiseret $O(1)$ tid.

	Ja	Nej	
$N - n$	A	B	24.1
$2n - N$	A	B	24.2
$\max\{0, 2n - N\}$	A	B	24.3
$n + \max\{0, 2n - N\}$	A	B	24.4
$\max\{0, 3n - N\}$	A	B	24.5
$2n + \max\{0, n - N\}$	A	B	24.6