EKSAMEN

Algoritmer og Datastrukturer

Onsdag 25. maj 2022, 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	
$\log n$ er $O(\sqrt{n})$	A	B	1.1
$\sqrt{n} \cdot \log n$ er $O(\sqrt{n})$	A	B	1.2
$n\sqrt{n}$ er $O(n^2 \log n)$	A	B	1.3
$(\log n)^3$ er $O(\log n)$	A	B	1.4
$n \cdot \log n$ er $O(n^{0.01})$	A	B	1.5
$(\log n)^6 + 2^{3 \log n}$ er $O(2^n)$	A	B	1.6
$\sqrt{n} + 5 \log(n^2)$ er $O(n^3)$	A	B	1.7
$2^{2 \log n}$ er $O(n \cdot \log n)$	A	B	1.8
$n!$ er $O(n \cdot \log n)$	A	B	1.9
$2^{\log n}$ er $\Theta(n)$	A	B	1.10
$n^{0.001}$ er $\Omega(n^2)$	A	B	1.11
$n\sqrt{n}$ er $\Theta(n^{3/2})$	A	B	1.12

Opgave 2 (Analyse af løkker, 6 %)

```
Algoritme loop1(n)
  s = 0
  for i = 1 to n
      for j = 1 to n
          for k = 1 to n
              s = s + 1
```

```
Algoritme loop2(n)
  i = n
  while i > 0
      i = i − 1
```

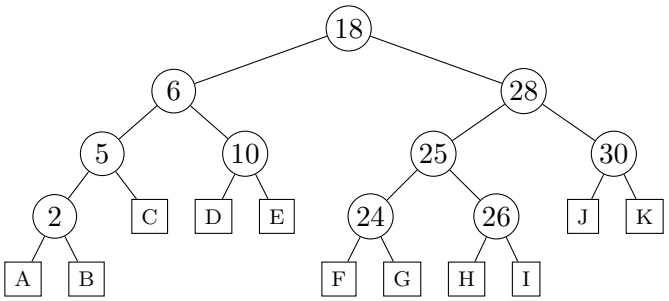
```
Algoritme loop3(n)
  s = 0
  for i = 1 to n
      for j = i to n
          for k = i to j
              s = s + 1
```

```
Algoritme loop4(n)
  i = 1
  while i ≤ n
      j = i
      while j > 0
          j = ⌊j/2⌋
      i = i + i
```

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

	$\Theta(n^2)$	$\Theta(n^2 \cdot \log n)$	$\Theta((\log n)^2)$	$\Theta(n \log n)$	$\Theta(\log n)$	$\Theta(n^3)$	$\Theta(\frac{\log n}{\log \log n})$	$\Theta(n)$	
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 8, 23, 14, 31 og 9 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(8)	A	B	C	D	E	F	G	H	I	J	K	3.1
INSERT(23)	A	B	C	D	E	F	G	H	I	J	K	3.2
INSERT(14)	A	B	C	D	E	F	G	H	I	J	K	3.3
INSERT(31)	A	B	C	D	E	F	G	H	I	J	K	3.4
INSERT(9)	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 6, 8, 13, 12, 7, 4 og 14 i den givne rækkefølge med MAX-HEAP-INSERT, startende med den tomme heap.

1	2	3	4	5	6	7	
14	12	13	6	7	4	8	A

1	2	3	4	5	6	7	
14	12	13	8	7	4	6	B

1	2	3	4	5	6	7	
14	13	12	8	7	6	4	C

1	2	3	4	5	6	7	
6	8	13	12	7	4	14	D

1	2	3	4	5	6	7	
13	12	14	8	7	4	6	E

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

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

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

Opgave 6 (Heap-Extract-Max, 4 %)

1	2	3	4	5	6	7	8	9	10	11	12	13
26	25	18	15	9	17	4	7	2	1	5	14	16

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

1	2	3	4	5	6	7	8	9	10	11	12	13	
25	18	16	9	17	15	7	2	1	5	14	4		A
25	15	18	7	9	17	4	2	1	5	14	16		B
25	15	18	7	9	17	4	16	2	1	5	14		C
25	16	18	15	9	17	4	7	2	1	5	14		D
25	15	18	7	9	17	4		2	1	5	14	16	E

Opgave 7 (Partition, 4 %)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	3	29	23	9	20	28	16	5	8	15	30	1	27	2

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	
11	3	29	1	23	9	20	28	16	5	8	15	30	27	2	A
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
11	3	29	1	5	8	9	15	16	20	23	28	30	27	2	B
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	2	3	5	8	9	11	15	16	20	23	27	28	29	30	C
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
11	3	29	1	9	20	28	16	5	8	15	30	23	27	2	D

Opgave 8 (Radix-sort, 4 %)

1222 1122 0403 4410 4403 1322

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.

0403	1122	1222	1322	4410	4403	A
0403	4403	4410	1122	1222	1322	B
4410	1222	1122	1322	0403	4403	C
0403	1122	1222	1322	4403	4410	D
0403	4403	4410	1222	1122	1322	E

Opgave 9 (Lineær probing, 4 %)

0	1	2	3	4	5	6	7	8	9	10
		19			9			10	14	21

I ovenstående hashtabel af størrelse 11 er anvendt *linear probing* med hashfunktionen $h(k) = 3k \bmod 11$.

Angiv positionerne de fem elementer 1, 2, 3, 4 og 7 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 9, 10, 14, 19 og 21).

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

Opgave 10 (Dobbelt hashing, 4 %)

0	1	2	3	4	5	6	7	8	9	10
22			12					1	3	18

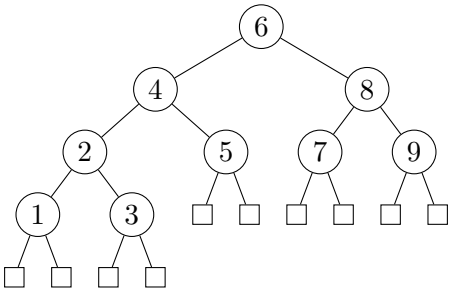
I ovenstående hashtabel af størrelse 11 er anvendt *dobbelt hashing* med hashfunktionerne $h_1(k) = 3k \bmod 11$ og $h_2(k) = 1 + (4k \bmod 10)$.

Angiv positionerne de fem elementer 5, 7, 8, 9 og 11 vil blive indsat på i hashtabellen (for hver af indsættelserne antager vi at hashtabellen kun indeholder elementerne 1, 3, 12, 18 og 22).

	0	1	2	3	4	5	6	7	8	9	10	
INSERT(5)	A	B	C	D	E	F	G	H	I	J	K	10.1
INSERT(7)	A	B	C	D	E	F	G	H	I	J	K	10.2
INSERT(8)	A	B	C	D	E	F	G	H	I	J	K	10.3
INSERT(9)	A	B	C	D	E	F	G	H	I	J	K	10.4
INSERT(11)	A	B	C	D	E	F	G	H	I	J	K	10.5

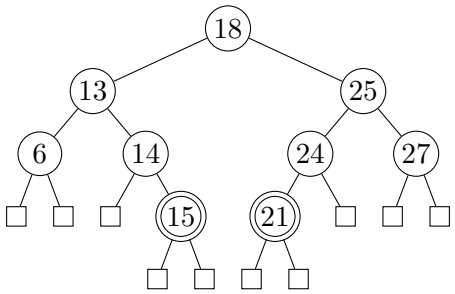
Opgave 11 (Rød-sort træ, 4 %)

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

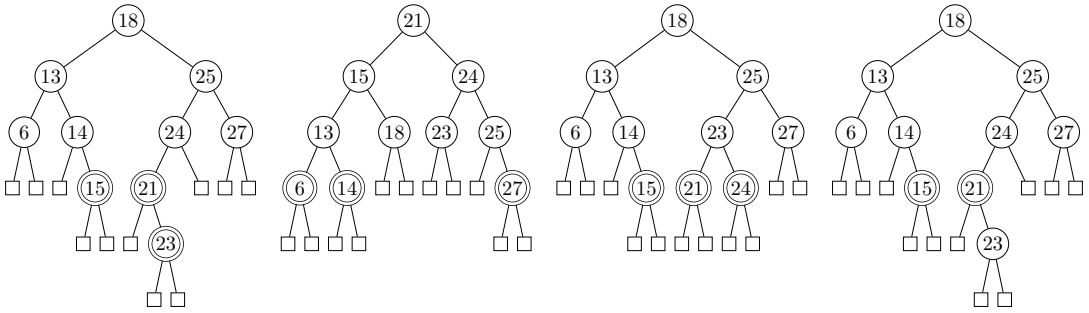


	Ja	Nej	
1, 2, 3, 5, 8	A	B	11.1
1, 3, 6	A	B	11.2
1, 3, 4, 8	A	B	11.3
1, 3	A	B	11.4
1, 3, 4, 7, 9	A	B	11.5

Opgave 12 (Indsættelse i rød-sort træer, 4 %)



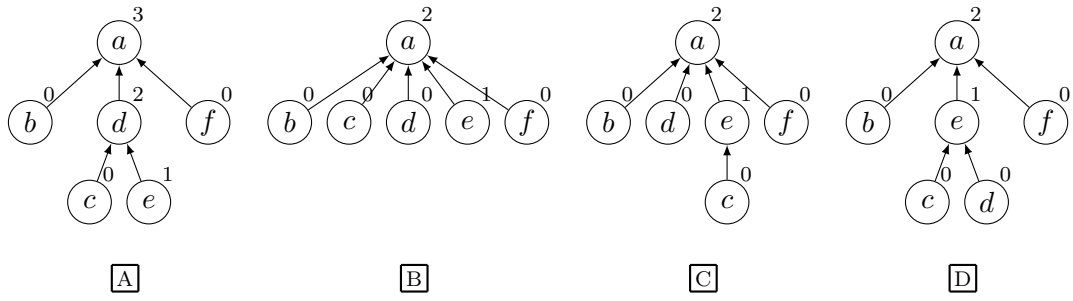
Angiv det resulterende rød-sorte træ når man indsætter 23 i ovenstående rød-sorte træ (dobbeltcirkler angiver røde knuder).



Opgave 13 (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(c, e)
UNION(c, d)
UNION(f, a)
UNION(c, a)
UNION(b, d)
FIND-SET(b)

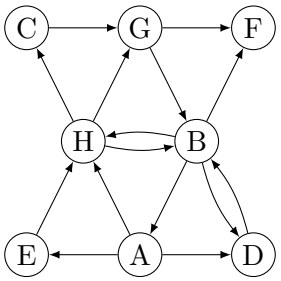


Opgave 14 (Rekursionsligninger, 4 %)

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

	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^3)$	
$T(n) = 4 \cdot T(n/2) + n^2$	A	B	C	D	E	F	G	14.1
$T(n) = 4 \cdot T(n/5) + n$	A	B	C	D	E	F	G	14.2
$T(n) = T(n-1) + n$	A	B	C	D	E	F	G	14.3
$T(n) = T(n/5) + 5$	A	B	C	D	E	F	G	14.4
$T(n) = 4 \cdot T(n/5) + n^3$	A	B	C	D	E	F	G	14.5

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**.

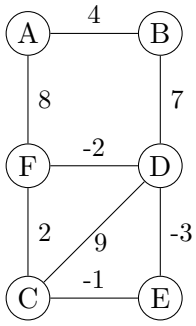
F D B G C H E A F G C B H E D A E G C H F B D A F G C H B D E A

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

Opgave 16 (Kruskals algoritme, 4 %)

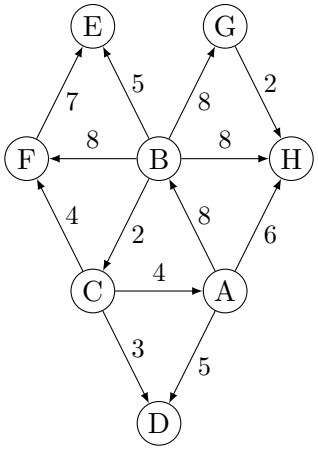


Antag Kruskals algoritme anvendes til at finde et minimum udspændende træ for ovenstående graf. Angiv hvilken kant **sidst** inkluderes i det mininmum udspændende træ.

(D , E) (C , E) (D , F) (C , F) (B , D)

A B C D E

Opgave 17 (Dijkstras algoritme, 4 %)

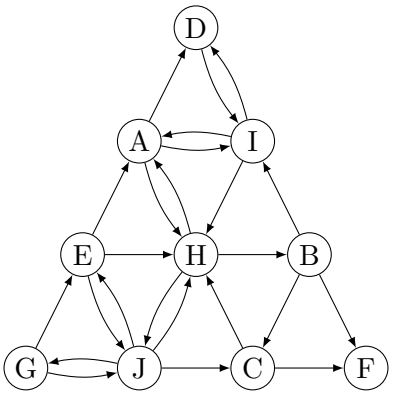


Antag Dijkstras algoritme anvendes til at finde korteste afstande fra **knuden A** til alle knuder i ovenstående graf. Angiv hvilken rækkefølge knuderne bliver taget ud af prioritetskøen i Dijkstra's algoritme.

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

A B C D

Opgave 18 (BFS, 4 %)

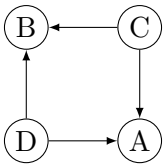


For et bredde først gennemløb (BFS) af ovenstående graf **startende i knuden A**, angiv rækkefølgen knuderne bliver indsat i køen Q i BFS-algoritmen. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

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

A B C D

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

Opgave 20 (Amortiseret analyse, 4 %)

Et rød-sort søgetræ understøtter INSERT(x) og DELETE(x), der henholdsvis indsætter et element x i et rød-sort søgetræ og sletter et element x fra et rød-sort søgetræ, begge i worst-case $O(\log n)$ tid, hvor n er den aktuelle størrelse af det rød-sorte søgetræ. Vi antager i denne opgave at x altid findes i det rød-sorte træ når vi udfører DELETE(x), dvs. der fjernes et element fra det rød-sorte træ.

Angiv for hver af nedenstående funktioner om den er en potentialefunktion, hvormed man kan argumentere for at INSERT tager amortiseret $O(\log n)$ tid og DELETE tager amortiseret $O(1)$ tid.

	Ja	Nej	
$\log n$	A	B	20.1
n	A	B	20.2
$\sum_{i=1}^{n} \log i$	A	B	20.3
$n \cdot \log n$	A	B	20.4
n^2	A	B	20.5

Opgave 21 (Invarianter, 4 %)

Nedenstående algoritme Multiplikation beregner produktet $n \cdot m$ af to heltal n og m, der kan være positive, negative og nul.

Algoritme Multiplikation(n, m)
Inputbetingelse : Heltal n og m
Outputkrav : $r = n_0 \cdot m_0$
Metode : $r \leftarrow 0$
$\{I\}$ **while** $n \neq 0$ **do**
if $n < 0$ **then**
$r \leftarrow r - m$
$n \leftarrow n + 1$
else
$r \leftarrow r + m$
$n \leftarrow n - 1$

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

	Ja	Nej	
$\lvert n \rvert = \lvert n \rvert - 1$	A	B	21.1
$\lvert n \rvert \leq \lvert n_0 \rvert$	A	B	21.2
$r = m \cdot n$	A	B	21.3
$n_0 \cdot m_0 = r + n \cdot m$	A	B	21.4
$n \cdot m = r + n_0 \cdot m_0$	A	B	21.5

Dynamisk programmering

De næste tre opgaver vedrører at løse det *begrænsede møntopdelings* problem ved hjælp af dynamisk programmering. Lad $C = \{c_1, c_2, \ldots, c_n\}$ betegne n forskellig positive og heltallige møntværdier. Vi ønsker at afgøre om en heltallig værdi V kan opnås med højest M mønter (den samme møntværdi må gerne indgå flere gange). F.eks. for møntværdierne $C = \{2, 5, 10\}$ kan værdien $V = 19$ opnås for $M = 5$, da $19 = 2 + 2 + 5 + 10$, men ikke for $M = 3$, da 19 ikke er summen af tre mønter.

For $0 \leq v \leq V$ og $0 \leq m \leq M$, lader vi $P(v, m)$ angive om det er muligt at opnå værdien v med højest m mønter. $P(v, m)$ kan bestemmes ved følgende rekursionsformel.

$$P(v,m) = \begin{cases} \text{Sand} & \text{hvis } v = 0 \\ \text{Falsk} & \text{hvis } v > 0 \text{ og } m = 0 \\ \text{Sand} & \text{hvis } m > 0 \text{ og der findes en mønt } c_i \leq v \text{ hvor } P(v - c_i, m - 1) \text{ er Sand} \\ \text{Falsk} & \text{ellers} \end{cases}$$

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

Algoritme Opdeling($V, M, (c_1, \ldots, c_n)$)
  Opret tabel $P[0\,..\,V\,,\, 0\,..\,M]$ hvor alle indgange er Falsk
  for ...
    for ... ≪ **Opgave 22**: iterer over P ≫
      ≪ **Opgave 23**: beregn $P[v, m]$ ≫
  if $P[V, M] =$ Sand **then**
    ≪ **Opgave 24**: Udskriv en løsning med højest M mønter ≫
  else
    print *ikke muligt*

Opgave 22 (4 %)

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

	Ja	Nej	
for $v = 0$ **to** V **for** $m = 0$ **to** M	A	B	22.1
for $v = 0$ **to** V **for** $m = M$ **to** 0 **step** - 1	A	B	22.2
for $v = V$ **to** 0 **step** - 1 **for** $m = 0$ **to** M	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

```
if v = 0 then
  P[v, m] = Sand
for i = 1 to n
  if P[v − c_i, m − 1] = Sand then
    P[v, m] = Sand
```

A B 23.1

```
if v = 0 then
  P[v, m] = Sand
for i = 1 to n
  if c_i ≤ v and P[v − c_i, m − 1] = Sand then
    P[v, m] = Sand
```

A B 23.2

```
if v = 0 then
  P[v, m] = Sand
if m > 0 then
  for i = 1 to n
    if c_i ≤ v and P[v − c_i, m − 1] = Sand then
      P[v, m] = Sand
```

A B 23.3

Opgave 24 (4 %)

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

Ja Nej

```
while V > 0 do
  for i = 1 to n
    if c_i ≤ V and P[V − c_i, M − 1] = Sand then
      print c_i
      V = V − c_i
      M = M − 1
```

A B 24.1

```
for i = 1 to n
  while c_i ≤ V and P[V − c_i, M − 1] = Sand do
      print c_i
      V = V − c_i
      M = M − 1
```

A B 24.2

```
for i = 1 to n
  if c_i ≤ V and P[V − c_i, M − 1] = Sand then
      print c_i
      V = V − c_i
      M = M − 1
```

A B 24.3