----------LÖSNINGSFÖRSLAG----------

Tentamen - Datastrukturer, algoritmer och programkonstruktion.
DVA104
Akademin för innovation, design och teknik
Onsdag 2018-08-15

Skrivtid: 08.30-13.30
Hjälpmedel: Handskrivna anteckningar (obegränsad mängd) samt ordbok/lexikon
Lärare: Caroline Uppsäll (anträffbar på 0704616110)

Preliminära betygsgränser

Betyg 3: 17p
Betyg 4: 25p
Betyg 5: 29,5p
Max: 33p

Allmänt

- Kod skriven i tentauppgifterna är skriven i C-kod.
- På uppgifter där du ska skriva kod ska koden skrivas i C.
- Markera tydligt vilken uppgift ditt svar avser.
- Skriv bara på ena sidan av pappret.
- Referera inte mellan olika svar.
- Om du är osäker på vad som avses i någon fråga, skriv då vad du gör för antagande.
- Oläsliga/Oförståeliga svar rättas inte.
- Kommentera din kod!
- Tips: Läs igenom hela tentan innan du börjar skriva för att veta hur du ska disponera din tid.

Lycka till!

Uppgift 1: *(1p)*

Vad innebär det att ett binärt sökträd är fullt?

a) Att det är sorterat
b) Att det är balanserat
c) Att det är minst 3 nivåer
d) Att varje nivå är fylld
e) Att en nod kan ha fler än 2 barn

Uppgift 2: *(1p)*

Divide and conquer är…

a) … en sorteringsalgoritm
b) … en datatyp¨
c) … en problemlösningsstrategi
d) … en sökalgoritm
e) … ett mått på effektivitet

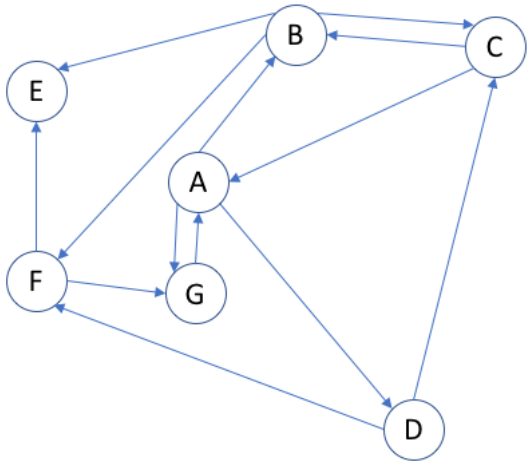
Uppgift 3: *(1p)*

Vilken/vilka av följande algoritmer är inte naturlig i sin effektivaste (en optimerad) implementation?

a) Bubbelsortering
b) Insättningssortering (Insertion)
c) Urvalssortering (Selection)
d) Merge sort
e) Quick sort

Det är ingen idé att chansa – felaktigt angivet svar ger poängavdrag på uppgiften. Totalpoängen på uppgiften kan dock bli lägst 0p. [varje felaktigt alternativt -0,5p]

Uppgift 4: *(2p)*
Antag nedanstående graf. Vilket/vilka av nedanstående alternativ är sanna för grafen?



a) Den är riktad
b) Den är viktad
c) Den är starkt sammanhängande
d) Den är sammanhängande
e) Nod A har grad 5 //även denna ger poäng
f) Det är en multigraf
g) Den innehåller inga cykler
h) Den är oriktad
i) Nod A har ingrad 2 och utgrad 3

Det är ingen idé att chansa – felaktigt angivet svar ger poängavdrag på uppgiften.
Totalpoängen på uppgiften kan dock bli lägst 0p. [varje felaktigt alternativt -0,5p]

Uppgift 5: *(4p)*
Skriv en rekursiv funktion som utför binärsökning på en linjär, sorterad mängd av heltal. ¨¨¨
Funktionen ska returnera 1 om det eftersökta datat hittas, annars 0.

Du kan anta att den array som skickas in till funktionen är sorterad samt att den innehåller minst ett element.

Funktionen ska vara rekursiv.

Följande funktionshuvud ska användas:

```
int binSearch(int arr[], int value, int low, int high)
```

Lösningsförslag

```
int binSearch(int arr[], int value, int low, int high)
{
        if(low <= high) – stop + return 0 [1p]
        {
                int mid = (low + high)/2; - hitta mitten [0,5p]
                if(arr[mid] == value) – hittat rätt + return 1 [0.5p]
                        return 1;
                else if(arr[mid] > value) – korrekt villkor [0,5p]
                        return binSearch(arr, value, low, mid-1); - korrekt anrop [0,5p]
                else
                        return binSearch(arr, value, mid+1, high); - korrekt anrop [0,5p]
        } returnera rekursiva anropet [0,5p]
        return 0;
}
```

Uppgift 6: *(3p)*
Vad skrivs ut?
Hur ser stacken R samt kön Q ut när programmet avslutas?

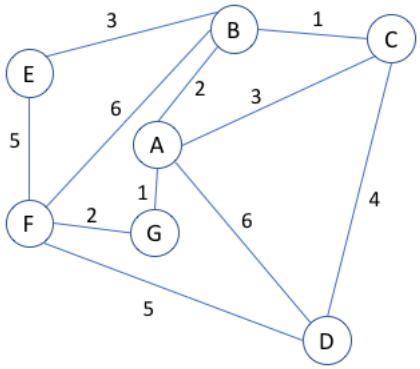
Typen Stack är en LIFO-kö (alltså en vanlig stack) och typen Queue är cirkulär.

Programmet nedan är skrivet i pseudokod.

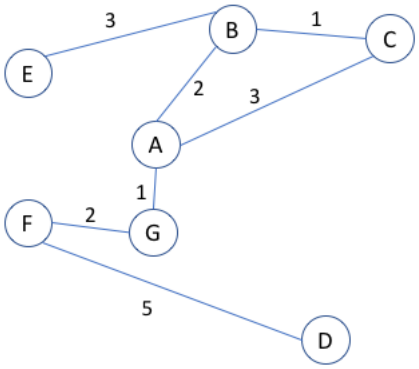
```
Stack R, S                          //stackarna R och S skapas
Queue Q(5)                          //kön Q med 5 platser skapas
Integer i = 1
While i < 60 do
      S.push(i)
      i = i + 10
end do
While !S.isEmpty() do               //så länge S inte är tom
      Q.enqueue(S.pop())
      If !S.isEmpty() then
            R.push(S.pop)
      End then
End do
Integer n = R.Length()              //Length() returnerar storleken på
stacken
Integer x = 0;
Q.enqueue(5)
While x < n do
      Print(Q.dequeue())            //Print skriver ut i konsollen
      Q.enqueue(R.peek()+x)
      x = x + 1
end do
```

Lösningsförslag
R innehåller (1p) top – 1 - 21 – 41
Q innehåller (1p) front – 2 – 3 – tom – 5 – 1 – back
Utskriften blir (1p) 51 31 11

Uppgift 7: *(3p)*
Antag följande graf

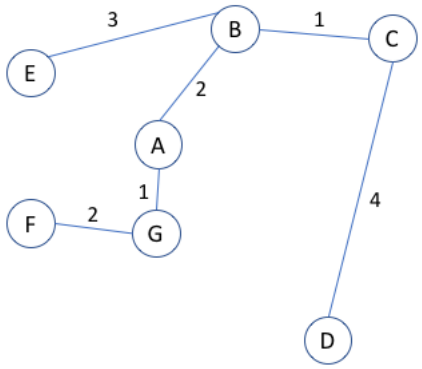


Från denna graf har följande påstådda minimum spanning tree med start i E tagits fram.



Är det påstådda minimum spanning tree korrekt? – Motivera ditt svar med max 5 meningar

Lösningsförslag
Svar Nej – det är inte korrekt. Ett korrekt MSP ser ut som följer



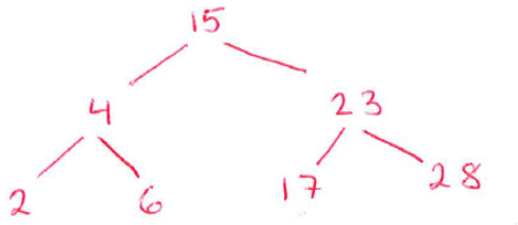
Man väljer hela tiden den kortaste möjliga vägen till en nod som inte redan besökts.

Motiveringens korrekthet bedöms för poäng.

Uppgift 8: *(2p)*

Ett och samma binära sökträd har skrivits ut med ordningarna Preorder(LR), Inorder(LR) och Postorder(LR). Vilket av nedanstående alternativ är utskriven med vilken traverseringsordning?

a) 4, 6, 15, 2, 23, 28, 17
b) 2, 4, 6, 15, 17, 23, 28 Inorder [0,5p]
c) 15, 4, 2, 6, 23, 17, 28 Preorder [0,5p]
d) 15, 4, 23, 2, 6, 17, 28
e) 2, 6, 4, 15, 17, 28, 23
f) 2, 6, 4, 17, 28, 23, 15 Postorder [1p]



Det är ingen idé att chansa – felaktigt angivet svar ger poängavdrag på uppgiften. Totalpoängen på uppgiften kan dock bli lägst 0p. [varje felaktigt alternativt -0,5p]

Uppgift 9: *(3p)*
Antag att du har en cirkulär arraybaserad kö (FIFO) enligt nedan.

```
int queue[10] = {};
int front = 0;
int back = 0;
```

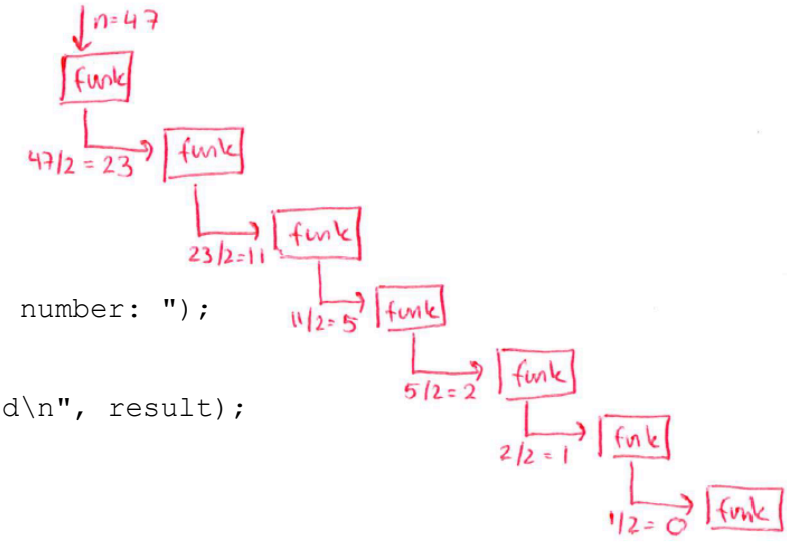
Din uppgift är skriva funktionen som tar bort ett värde ur kön (dequeue). Du ska använda dig av funktionshuvudet:

```
int dequeue(int queue[], int front, int back, int arrSize)
```

Lösningsförslag – för poäng bedöms lösningen som helhet

```
int dequeue(int queue[], int front, int back, int arrSize)
{
        if(front == back)
                return -1;          //kön är tom
        front = (front + 1) % arrSize;
        return front;
}
```

Uppgift 10: *(3p)*
Antag nedanstående program




```
#include <stdio.h>

int funk(int n);

int main()
{
      int num, result;
      printf("Enter a decimal number: ");
      scanf("%d", &num);
      result = funk(num);
      printf("The result is %d\n", result);
}

int funk(int n)
{
      if (n == 0)
            return 0;
      else
            return (n % 2) + 10 * (funk(n / 2));
}
```

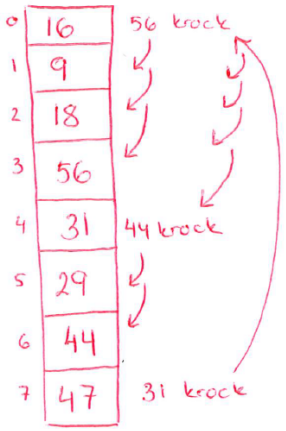
a) Hur många anrop görs till funk om num är 47? – beskriv gärna svaret med bild/text [1p] – svar 7 anrop
b) Vilken komplexitet har programmet med avseende på n, svara i termer av Ordo – motivera med max 3 meningar [1p] – (O(log n)) mängden halveras för varje anrop
c) Finns det något logiskt problem med ovanstående program – t.ex. någonting som gör att programmet krashar (förutom att programmet inte kontrollerar att indatat verkligen är ett heltal)? – Motivera med max 3 meningar [1p] – NEJ, funktionen hanterar alla typer av heltal (även negativa) – oavsett input kommer basfallet att nås.

Uppgift 11: *(3p)*
Antag att du har en hashtabell som löser krockar med öppen adressering (linjär sondering). Tabellen har 8 platser och är från början tom. Hashfunktionen gör nyckel % 8.

a) Visa (med bild eller text) hur tabellen ser ut när följande nycklar (i angiven ordning) läggs till, visa också beräkningen av indexet.[1p]

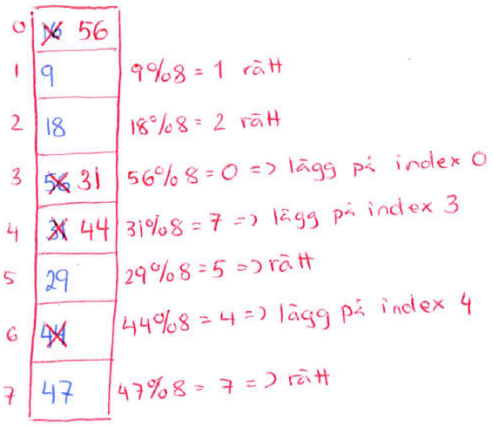
Nycklar: 9, 16, 18, 29, 56, 47, 31, 44
%8: 1 0 2 5 0 7 7 4

Lösning:
Index 0 – 16
Index 1 – 9
Index 2 – 18
Index 3 – 56
Index 4 – 31
Index 5 – 29
Index 6 – 44
Index 7 – 47



b) Utgå från hashtabellen du skapade i a). Visa hur tabellen ser ut efter att nyckeln 16 tagits bort [2p]

Lösning:
Index 0 – 56
Index 1 – 9
Index 2 – 18
Index 3 – 31
Index 4 – 44
Index 5 – 29
Index 6 – tom
Index 7 – 47

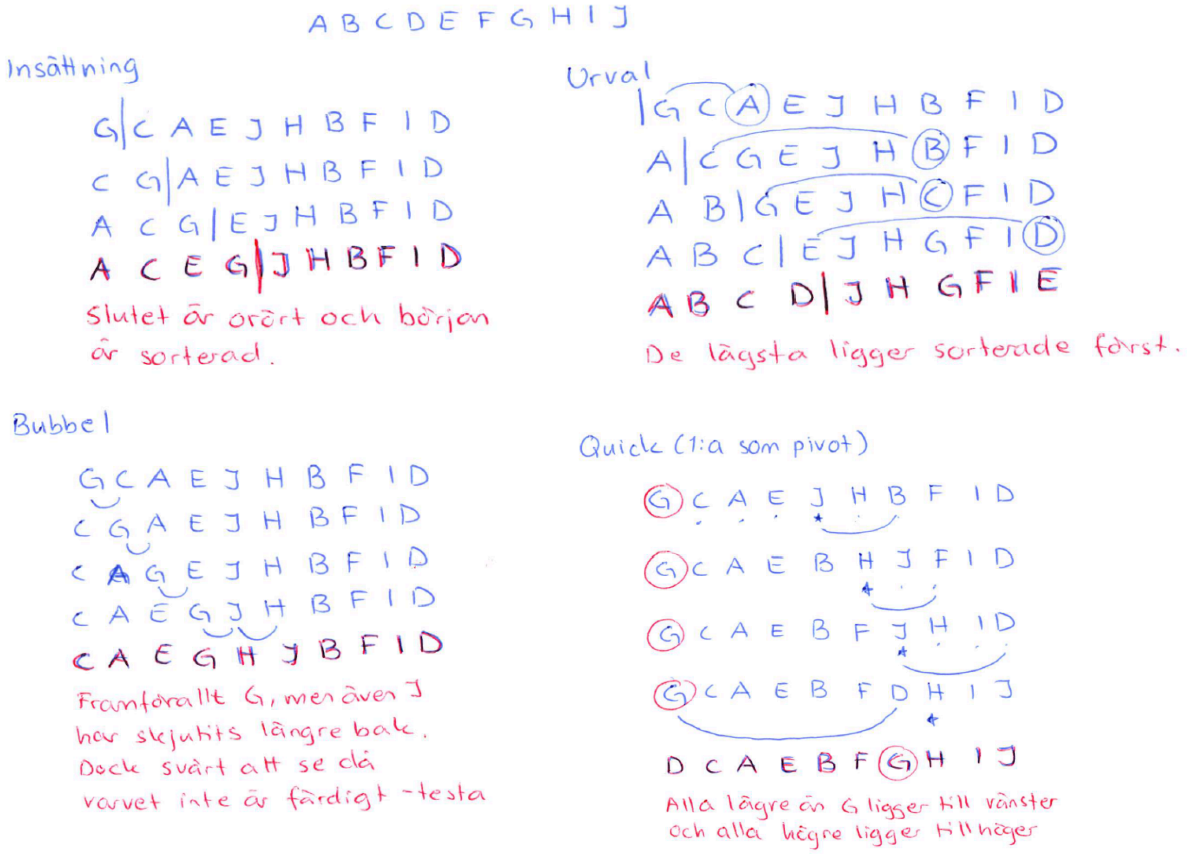


Alla efterföljande måste tas bort och sedan läggas till igen för att upprätthålla hash-reglerna vid sökning.

Uppgift 12: *(4p)*
Följande mängd har sorterats (i bokstavsordning) mha några av algoritmerna Bubbelsortering, Insättninssortering (Insertion), Urvalssortering (Selection), Merge sort och Quick sort (1:a elementet vald som pivot). a-f nedan är någonstans i dessa algoritmers sortering. Det kan finnas alternativ som inte är någon av de nämna algoritmerna och det är inte säkert att alla algoritmer finns representerade.

Originalmängd: G C A E J H B F I D

a) D C A E B F G H I J Quick
b) A C E G J H B F I D Insättning
c) A B E D C F G H I J
d) A B C D J H G F I E Urval
e) A C E G B H F I D J
f) C A E G H J B F I D Bubbel



Uppgift 13: *(3p)*
Antag att vi har implementerat vårt binära sökträd på följande vis:

```
struct treenode
{
	struct treenode* left;
	struct treenode* right;
	int data;
};
typedef struct treenode* BSTree;
```

Skriv nu en funktion som söker i trädet. Använd följande funktionshuvud. Funktionen ska returnera 1 om datat hittas och annars 0

```
int search(BSTree tree, int data)
```

Lösningsförslag – för poäng/delpoäng görs en bedömning av lösningens korrekthet

```
int search(BSTree tree, int data)
{
	if(tree == NULL)
		return 0;
	if(tree->data == data)
		return 1;
	else
	{
		if(data < tree->data)
			return search(tree->left, data);
		else
			return search(tree->right, data);
	}
}
```