Norsk informatikkolympiade 2014–2015 — 1. runde

Sponset av

UNIVERSITETET I BERGEN
Institutt for informatikk

CFEngine FFI Forsvarets forskningsinstitutt bouvet

Uke 46, 2014

Tid: 90 minutter
Tillatte hjelpemidler: Kun skrivesaker. Det er *ikke* tillatt med kalkulator eller trykte eller håndskrevne hjelpemidler.
Instruksjoner: Oppgavesettet består av 16 oppgaver, med fire svaralternativer på hver oppgave. Det er kun ett riktig svar på hver oppgave. Du får fire poeng for hvert riktige svar, null poeng for feil svar, og ett poeng for hver oppgave du ikke svarer på (det vil si at det ikke lønner seg å gjette dersom du ikke vet hva svaret er). Du kan godt krysse av på alternativene i oppgaveteksten underveis, men du *må* føre inn svarene på svararket helt bakerst.
Oppgavene som handler om programmering starter med beskrivelser av temaet de handler om, slik at de som ikke har vært borti programmering før også kan prøve seg på disse oppgavene. De som kan programmere trenger ikke å lese disse beskrivelsene; all koden oppfører seg som i vanlige programmeringsspråk.

FASIT

1. En toglinje går mellom 10 stasjoner som alle ligger på en rett linje med 1 kilometers avstand mellom nabostasjoner. Du ønsker å reise fra den ene endeholdeplassen til den andre og tilbake igjen. Hvor mange kilometer har du da reist?

   A. 10

   B. 18

   C. 19

   D. 20

   Løsningskommentar: Selve toglinjen er bare 9 km lang. Du reiser den 2 ganger, altså 18 km.

2. Vi har konstruert en robot som beveger seg basert på følgende tre kommandoer

   `A`: Gå frem 1 meter. Sving deretter 90 grader til venstre på stedet.

   `B`: Gå frem 2 meter. Sving deretter 90 grader til høyre på stedet.

   `C`: Gå 1 meter bakover (uten å snu seg).

   Et *program* består av en kjede med kommandoer som roboten skal utføre i rekkefølge. Til å begynne med står roboten på en uendelig stor slette. Hvilket av følgende programmer kan man gi til roboten for at den skal ende opp på samme sted som den begynte?

   A. A B C C A C A

   B. B C C B B C A

   C. C A A C C B A

   D. A B A B C C B

3. Du har to funksjoner som fungerer på tekststrenger (dvs. bokstavsekvenser, eller *ord*). `LIM` tar to tekststrenger og limer dem sammen. F.eks. så gir kallet `LIM(fort,frem)` tekststrengen `fortfrem`. `SNU` tar en tekststreng og snur rekkefølgen av bokstavene i den. F.eks. så gir kallet `SNU(sakte)` tekststrengen `etkas`. Hvilket av de følgende sammensatte funksjonskall gir ordet "informatikk" ?

   A. `LIM(SNU(LIM(itam,infor)),kk)`

   B. `LIM(SNU(LIM(ofni,tamr)),ikk)`

   C. `LIM(SNU(LIM(tamro,fni)),ikk)`

   D. `LIM(SNU(LIM(amr,ofni)),kkit)`

4. Hvis du skal skrive alle tallene fra og med 1 til og med 2500, hvor mange ganger må du skrive sifferet 2?

   A. 1250

   B. 1251

   C. 1300

   D. 1301

Løsningskommentar: Tallene fra og med 2000 til og med 2500 (501 tall) har alle en 2'er i 1000-plassen. Tallene 200-299, 1200-1299 og 2200-2299 (300 tall) har alle 2-er på 100-plassen. 1/10 av tallene har en 2'er på 10-plassen, og 1/10 av tallene har en 2-er på 1-plassen. Totalt blir det da $501 + 300 + 250 + 250$ 2-ere.

5. I det binære tallsystemet så bruker man kun to forskjellige sifre: 0 og 1. Sifferet lengst til høyre er 1'er plassen, til venstre for dette har man 2'er-plassen, deretter 4'er plassen, 8'er plassen, 16-plassen, og så videre. Hvert siffer er altså dobbelt så mye verdt som sifferet det står til venstre for. Verdien av et tall i det binære tallsystemet finner man ved å summere verdiene for posisjonene som har sifferverdi 1. For eksempel så tilsvarer det binære tallet 11010 verdien $16 + 8 + 2$, altså 26. Hvordan skriver man tallet 102 i binært?

   A. 1100110

   B. 1101110

   C. 1011010

   D. 1011110

Løsningskommentar: $102 = 64 + 32 + 4 + 2$

6. På en datamaskin har man som regel bare et visst antall binære sifre (også kalt *bits*) til å uttrykke tall med. Hva er det største tallet man kan uttrykke med 8 binære sifre, hvis nøyaktig 4 av sifrene skal ha verdien 0? (Hvis du kan programmere fra før så snakker vi her om en `unsigned byte`.)

   A. 128

   B. 170

   C. 240

   D. 256

Løsningskommentar: Det største tallet man kan skrive er 11110000 som tilsvarer $128 + 64 + 32 + 16$.

7. En *bitsekvens* er en følge med 0'er og 1 ere. F.eks. så er 01100 en bitsekvens med lengde 5. Hvis k er antall bitsekvenser med lengde 10 som ikke inneholder to 1-tall etter hverandre, hva er da siste siffer i k?

   A. 2

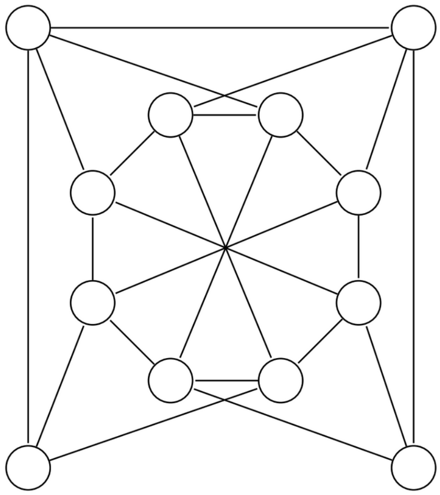
   B. 4

   C. 7

   D. 9

Løsningskommentar: Alle gyldige sekvens med lengde n kan dannes ved å enten ta en hvilken som helst gyldig sekvens med lengde $n-1$ og legge til 0 bak, eller en hvilken som helst sekvens av lengde $n-2$ og legge til 01 bak. Antall gyldige sekvenser med en gitt lengde danner derfor fibonacchi-tallfølgen, og vi har at det er 144 gyldige sekvenser med lengde 10.

8. I datasammenheng er en *graf* en samling av *noder* (punkter; her tegnet som sirkler) og *kanter* (streker) mellom nodene. Hvor mange forskjellige farger trenger man for å fargelegge grafen under, dersom man krever at for alle kanter så har de to nodene som kanten forbinder forskjellige farger?



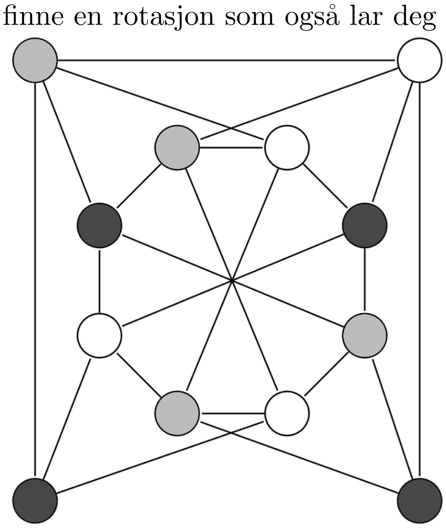
   A. 2

   B. 3

   C. 4

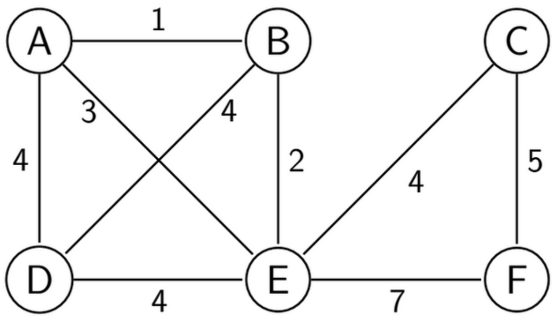
   D. 5

Løsningskommentar: Man ser fort at grafen ikke kan fargelegges med 2 farger. 8-kanten i midten er det som er vanskelig å få til å fargelegge, så det er lurt å begynne med å prøve å fargelegge den. Ved litt utprøving så ser man at den eneste måten å 3-farge den på er om fargene er A-B-C-A-B-C-B-C i rekkefølge rundt 8-kanten. Fra dette så er det lett å finne en rotasjon som også lar deg fargelegge de 4 hjørnene.



9. Grafen under representerer et veinettverk over noen byer (sirklene A - F) der linjene mellom byene representerer potensielle veier. Hver vei koster en viss sum å bygge ut. Disse kostnadene er tallene som står skrevet ved de potensielle veiene. Du ønsker å bygge veier slik at alle byene

blir forbundet med hverandre via en eller flere veier. Det er viktig å holde budsjettet, så du vil bygge veinettet slik at kostnaden blir minst mulig. Hva er minste mulig kostnad for et veinett som forbinder alle byene?



A. 14

B. 15

C. 16

D. 17

Løsningskommentar: Dette er et kjent informatikkproblem som kalles *minimum utspennende tre*. Dette kan løses ved at man alltid bygger den billigste veien som forbinder to byer som det ikke er mulig å nå fra hverandre. Altså vil man først bygge veien A-B. Deretter bygger man veien B-E. Veien A-E hopper man over siden det allerede er en forbindelse mellom A og E. Det er mange veier med kostnad 4 på, så rekkefølgen blir litt vilkårlig, men man må bygge veien E-C og en av veiene ut fra D. Deretter gjenstår bare C-F for å gjøre grafen sammenhengende.

10. I de fleste programmeringsspråk brukes operatoren `=` på en annen måte enn i matematikken. Den instruerer nemlig datamaskinen om å regne ut verdien av uttrykket på høyre side av likhetstegnet og legge den inn i *variabelen* på venstre side. (Verdien *kopieres* alltid, selv hvis høyresiden er en enkelt variabel — den *flyttes* ikke.) En variabel holder på en verdi helt frem til du legger noe annet inn i den; da forsvinner den gamle verdien. Linjer som står etter hverandre utføres etter tur. For eksempel vil

```
x = 4
x = x + 3
```

gjøre at variabelen `x` ender opp med å inneholde verdien 7. (I første linje så settes `x` til å ha verdien 4. I neste linje regnes $x + 3$ ut fra `x` sin daværende verdi, altså som $4 + 3 = 7$. Denne verdien lagres da i `x` og overskriver verdien 4.)

Hva blir verdien av `y` etter at følgende kode er utført?

```
x = 3
y = x + 2
x = x + 3
y = x * y
```

A. 5

B. 15

C. 20

D. 30

Løsningskommentar: Først settes x til å være 3. Detter settes y til å være 2 mer enn x, altså 5. Deretter settes x til å være 3 mer enn hva den var, alltså til å bli 6. Til sist settes y lik $6 * 5 = 30$.

11. Noen tall har den egenskapen at de kan skrives som en sum av etterfølgende positive heltall. For eksempel så er $24 = 7 + 8 + 9$ og $15 = 1 + 2 + 3 + 4 + 5$. Hva er lengden av den lengste sekvensen av etterfølgende tall som har summen 150?

    A. 12

    B. 13

    C. 14

    D. 15

Løsningskommentar: Summen av tallene fra 1 til n er $\frac{n^2+n}{2}$. Summen av tallene fra $m+1$ til n blir da $\frac{n^2+n}{2} - \frac{m^2+m}{2}$. Ved å lage en tabell over tallene $\frac{n^2+n}{2}$ for forskjellige verdier av n så ser vi at det første tallet som vi kommer til som er $\frac{m^2+m}{2}$ større enn 150 for et eller annet heltall m er $\frac{17^2+17}{2} = 150 + \frac{2^2+2}{2}$

12. I Merkeligstan så kommer pengene i veldig rare valører. Man har mynter med verdien 1, 4, 10, 21 og 50 kroner. Du har kjøpt noe som koster 88 kroner på markedet. Hva er det minste antallet mynter du må gi fra deg for å kunne betale dette beløpet?

    A. 3

    B. 4

    C. 5

    D. 6

Løsningskommentar: $88 = 21 + 21 + 21 + 21 + 4$

13. UKULT (Utrolig Kjedelige og Unaturlige LekeTøy) har produsert følgende spill for små barn. De har skrevet ut en haug papirbiter med bokstaven 'a' på, en haug med ordet 'bc' på, og en haug med ordet 'de' på. UKULT har lyst på å skrive noen artige fakta på pakken, blant annet hvor mange forskjellige ord bestående av 8 bokstaver man kan lage ved å sette sammen lapper. Merk at 'debcaabc' er et veldig fint ord på lengde 8 etter UKULT sine standarder; ordene trenger altså ikke å eksistere i noe ordentlig språk. Hvor mange slike ord finnes det av lengde 8?

A. 81

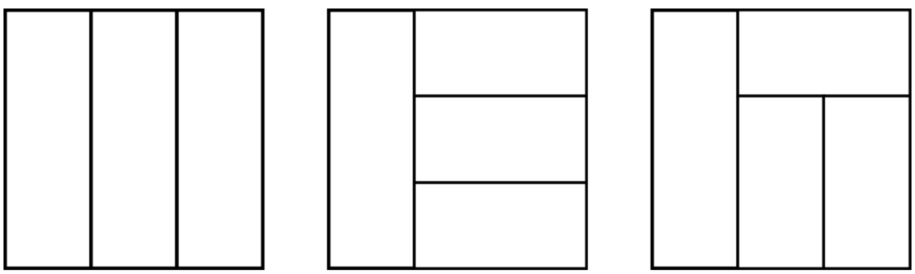
B. 111

C. 141

D. 171

Løsningskommentar: Antall ord på n bokstaver som kan skrives gis ved funksjonen $f(n) = f(n-1) + f(n-2) + f(n-2)$ når $n > 2$, $f(1) = 1$ og $f(2) = 3$.

14. På sagbruket like ved der NIO holder til har de en sag som de bruker til å kappe opp treplater til planker. Man sender inn en treplate med størrelse $n\,\text{dm} \times m\,\text{dm}$, der n og m begge er heltall større enn 1. ($1\,\text{dm} = 10\,\text{cm}$). Maskinen vil kutte av en planke med bredde $1\,\text{dm}$ fra den ene siden av treplaten, og man ender opp med en planke på lengde $m\,\text{dm}$ og en treplate på $(n-1)\,\text{dm} \times m\,\text{dm}$ (hvis $n-1 = 1$ regnes også denne platen for å være en planke). Merk at maskinen ikke kan kutte opp en treplanke i kortere planker. Siden man kan rotere treplaten mellom hver gang man skjærer av en planke, er det mange måter å dele den opp på. F.eks. kan man skjære opp en plate på $3\,\text{dm} \times 3\,\text{dm}$ på disse måtene:



Ettersom det bare er hvor mange planker av hver lengde som spiller noen rolle, regnes de siste to bildene ovenfor som samme oppdeling. Der er altså bare 2 måter å dele opp en $3\,\text{dm} \times 3\,\text{dm}$ plate til planker på: enten til 3 planker med lengde $3\,\text{dm}$, eller til 1 planke med lengde $3\,\text{dm}$, og 3 planker med lengde $2\,\text{dm}$. La m være antall måter å dele opp en plate på størrelse $6\,\text{dm} \times 6\,\text{dm}$. Hva er siste siffer i m?

A. 0

B. 2

C. 4

D. 8

Løsningskommentar: La $f(n, m)$ være antall måter å dele opp en plate på størrelse $ndm \times mdm$ i planker på. Dersom $n = 1$ eller $m = 1$ så er $f(n, m) = 1$ fordi platen allerede er en planke. Hvis $n \neq m$ så er $f(n, m) = f(n-1, m) + f(n, m-1)$ fordi man kan skjære av en planke fra enten langsiden eller kortsiden. Hvis $n = m$ spiller det ingen rolle hvilken side man skjærer fra så da er $f(n, m) = f(n-1, m)$. Ved å bygge opp en tabell der man har n på rader og m på kolonner og $f(n, m)$ i kryssningspunktene kan man fort regne seg til at $f(6, 6) = 42$

15. En *funksjon* er en navngitt samling med programinstruksjoner, som kan *kalles* (startes) med *inndata* og *returnere* (gi tilbake) *utdata*. Kommandoen `if` sjekker om uttrykket som står i parenteser er **true** (sann) eller **false**(usann); hvis det er **true**, gjøres det som står i krøllparentesene

etter `if`; hvis det er **false**, gjøres det som står i krøllparentesene etter den tilhørende `else`'n. `return` avslutter funksjonen og returnerer resultatet av uttrykket som står på samme linje. `<` er den vanlige mindre-enn-operatoren — f.eks. vil `3 < 4` bli **true**, og `4 < 3` blir **false**. `3 < 3` blir også **false**.

Funksjoner i programmeringsspråk kan også inneholde kall til seg selv. Dette kalles *rekursive* funksjoner. Når en rekursiv funksjon kaller seg selv, starter en ny utgave av den samme funksjonen, og den gamle utgaven venter til den nye er blitt ferdig. Gitt følgende definisjon av funksjonen `f`:

```
f(x,y,z) {
    if (x < y) {
        return 1 + f(x + 1, y, z)
    } else if (y < z) {
        return 2 + f(x, y + 1, z)
    } else {
        return z
    }
}
```

Hva returnerer funksjonskallet `f(12,14,20)`?

A. 20

B. 32

C. 40

D. 48

Løsningskommentar: Rekursjonen ender på f(20, 20, 20).

16. En *while-løkke* utfører koden inni krøllparentesene sine gjentatte ganger. Før hver utførelse sjekker løkken betingelsen inni parentesene rett etter `while`. Dersom betingelsen er usann, avsluttes løkken, og man fortsetter med koden etter avslutningskrøllparentesen (dersom det er noe kode der i det hele tatt).

Et *array* er en samling med et bestemt antall elementer. `A[0]` er det første elementet, `A[1]` er det andre osv. Antallet elementer totalt er `length(A)`. Det siste elementet er dermed `A[length(A) - 1]`.

Hva returnerer funksjonen under dersom vi kaller den med arrayet `{1, 3, 5, 7, 9, 11, 13}`?

```
mysterium(A) {
    j = 0
    s = 0
    while (j + 1 < length(A)) {
        if (A[j] < A[j + 1]) {
            s = s + A[j] + A[j + 1]
            k = A[j]
            A[j] = A[j + 1]
            A[j + 1] = k
```

```
            j = 0
        } else {
            j = j + 1
        }
    }
    return s
}
```

A. 294

B. 349

C. 720

D. 1029

Løsningskommentar: Funksjonen sorterer tallene i synkende rekkefølge ved å bytte om på tall som står ved siden av hverandre. Hver gang funksjonen bytter om på to tall så økes verdien av s med summen av de to tallene. Siden funksjonen kalles med 7 talll som alle står sortert i stigende rekkefølge, så må hvert tall på ett eller annet tidspunkt bytte plass med alle de andre. s vil da ende opp med verdien $(1 + 3 + 5 + 7 + 9 + 11 + 13) \times 6$

Svarark

Oppgave	**A**	**B**	**C**	**D**	Poeng (til bruk for læreren)
1		x			
2		x			
3			x		
4				x	
5	x				
6			x		
7		x			
8		x			
9			x		
10				x	
11				x	
12			x		
13				x	
14		x			
15			x		
16	x				
Poengsum					

Til læreren: Husk at korrekt svar gir 4 poeng, feil svar gir 0 poeng, og fraværende svar gir 1 poeng.