TÖL101G - Tölvunarfræði 1

Vikublað 7

Lausn

Cloudcoder Æfingar

2.3.2

Athugið að $\log(N!) = \log(N) + \log((N-1)!)$ og $\log(0!) = 0$. Ef við útfærum þetta endurkvæma fall beint fæst nákvæmari lausn sem að lendir ekki í vandræðum með stór N.

```
public class LogFact {
    // Notkun: x = logFact(N)
    // Fyrir: N >= 0
    // Eftir: x = log(N!)
    public static double lnFactorial(int N) {
        if (N == 0) {
            return 0.0;
        }
        return Math.log(N) + lnFactorial(N-1);
    }
}
```

2.1.19

Þar sem fylkið sem við tökum inn er með heiltölum getum við notað gildin sem vísa (e. index) á réttan stað í útkomufylkinu og hækkað rétta tölu um 1, t.d. með `b[a[i]]++`. Þá verðum við reyndar að passa okkur að vísa ekki út fyrir fylkið og athugum hvort `a[i]` sé á réttu bili.

```
public class Histogram {

  // Notkun: b = histogram(a,M)
  // Fyrir:  M >= 0
  // Eftir:  b er fylki af lengd M, þ.a. b[i] er
  //         fjöldi staka í a sem er jafn i
  public static int[] histogram(int[] a, int M) {
    int[] b = new int[M];
    for (int i = 0; i < a.length; i++) {
      // I: b[j] er fjöldi staka í a[0,..,i-1] sem er jafn j, fyrir j=0..M-1
      if (a[i] >= 0 && a[i] < M) {
        b[a[i]]++;
      }
    }
    return b;
  }
}
```

Æfingar

2.3.4

Gildið úr `ex234(6)` er ‘‘`311361142246`’’

2.3.8

Fallið mystery skilar margfeldi a og algildinu af b, þ.e. $a \cdot |b|$. Þegar öllum $+$ er breytt í * þá skilar fallið alltaf 0 því í grunntilfellinu er skilað 0 og 0 margfaldað með hverju sem er verður áfram 0. Ef grunntilvikinu er líka breytt í að skila 1 fæst fall sem skilar a í veldinu algildinu af b, þ.e. $a^{|b|}$.

1

Lausnin miðast við að nota fyrsta stak sem skiptistak, líkt og gert var á glærunum.

Hér er fyrst sýnt fylkið sjálft, og síðan fyrir hvert falla kall er fylkið prentað fyrir og eftir skiptingu. Að lokum er raðaða fylkið prentað út.

```
{6, 2, 1, -1, 2, 5, 1, 3, 2}
a[0, 8] = {6, 2, 1, -1, 2, 5, 1, 3, 2}
a[0, 8] = {2, 2, 1, -1, 2, 5, 1, 3, 6}
 a[0, 7] = {2, 2, 1, -1, 2, 5, 1, 3}
 a[0, 7] = {1, 1, -1, 2, 2, 5, 2, 3}
  a[0, 2] = {1, 1, -1}
  a[0, 2] = {-1, 1, 1}
   a[0, 0] = {-1}
   a[2, 2] = {1}
  a[4, 7] = {2, 5, 2, 3}
  a[4, 7] = {2, 5, 2, 3}
   a[4, 3] = {}
   a[5, 7] = {5, 2, 3}
   a[5, 7] = {3, 2, 5}
    a[5, 6] = {3, 2}
    a[5, 6] = {2, 3}
     a[5, 5] = {2}
     a[7, 6] = {}
    a[8, 7] = {}
 a[9, 8] = {}
{-1, 1, 1, 2, 2, 2, 3, 5, 6}
```

2

Athugið að við vitum að $F \to I$, $\{I \land R\}S\{I\}$ og $I \land \neg R \to E$ gilda. Fyrir mótdæmin munum við nota eftirfarandi forrit af vikublaði 4.

```
int k = 0, n = 1;
//F : N >=1,k=0,n=1
while (n <= N/2) {
  //I: n = 2^k og n <= N
  n *= 2;
  k++;
}
// E: 2^k ≤ N < 2^(k+1)
```

- Ef F gildir þá gildir I einnig þar sem $F \to I$. Svo ef $F \land R$ gildir þá hlýtur $I \land R$ að gilda einnig. Ef við keyrum forritsbútinn S þá vitum við að þar sem $\{I \land R\}S\{I\}$ er satt hlýtur I að gilda eftir. Því höfum við sýnt að $\{F \land R\}S\{I\}$ er alltaf satt.
- Hér þarf aðeins að finna eitt forrit sem uppfyllir öll 3 skilyrðin og eitt ástand þar sem I gildir, en ef við keyrum S þá gildir I ekki. Þar sem $\{I \land R\}S\{I\}$ gildir almennt sést að R má ekki vera satt, þ.e. við erum ekki enn í lykkjunni. Tökum

sem dæmi ástandið $k = 0, n = 1, N = 1$ sem uppfyllir I ef við keyrum S þá verður ástandið $k = 1, n = 2, N = 1$ sem uppfyllir ekki I.

- Þar sem $I \wedge \neg R \rightarrow E$ gildir sést að til að fá mótsögn þurfum við að finna ástand þar sem R gildir. T.d. upphafsástandið með $k = 0, n = 1, N = 4$ uppfyllir I en ekki E og því gildir yrðingin $I \rightarrow E$ ekki.