PAL: 12. cvičení

Tomáš Sieger

10. 12. 2020

Opakování z minula

Př. 10/2: náhodná čísla

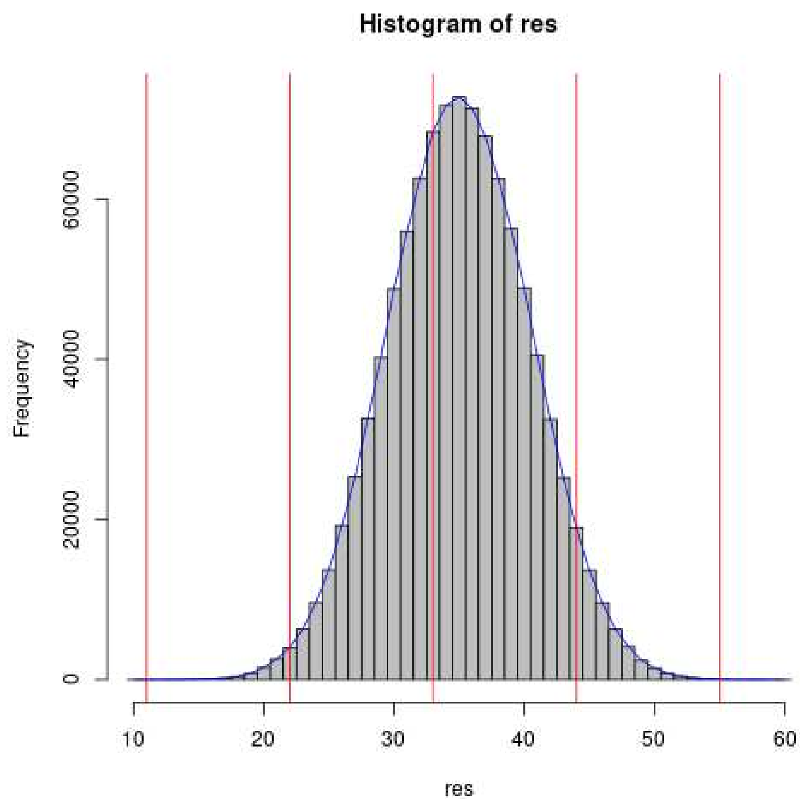
Máte jednu hrací kostku. Popište, jak využijte házení kostkou tak, abyste měli generátor náhodných celých čísel v rozmezí $0 \dots 10$. Všechna čísla $0, 1, 2, \dots 10$ musí být generována se stejnou pravděpodobností.

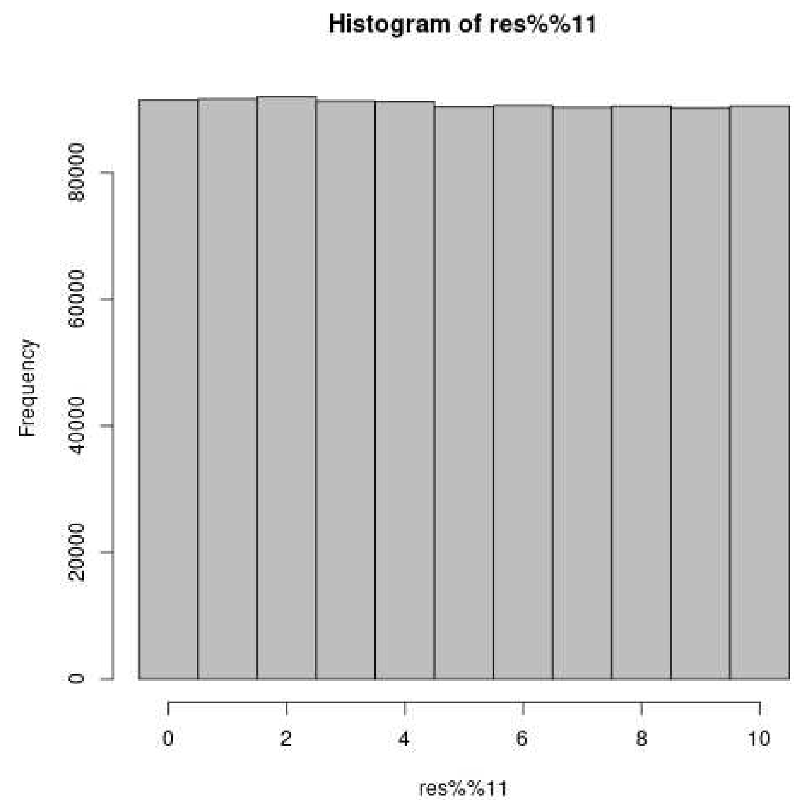
Podle dopadu kostky, kam míří jeden roh (rozdělím prostor na 11 dílů)

↳ "jako kostky"

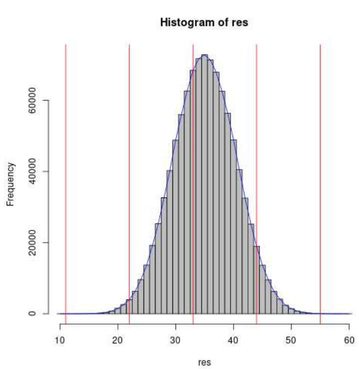
Hodím 10x kostkou, hodnoty sečtu modulo 11.

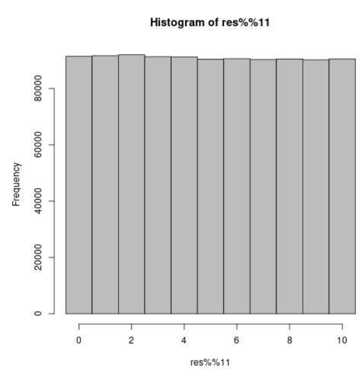
Hodím 10x kostkou, hodnoty sečtu modulo 11.

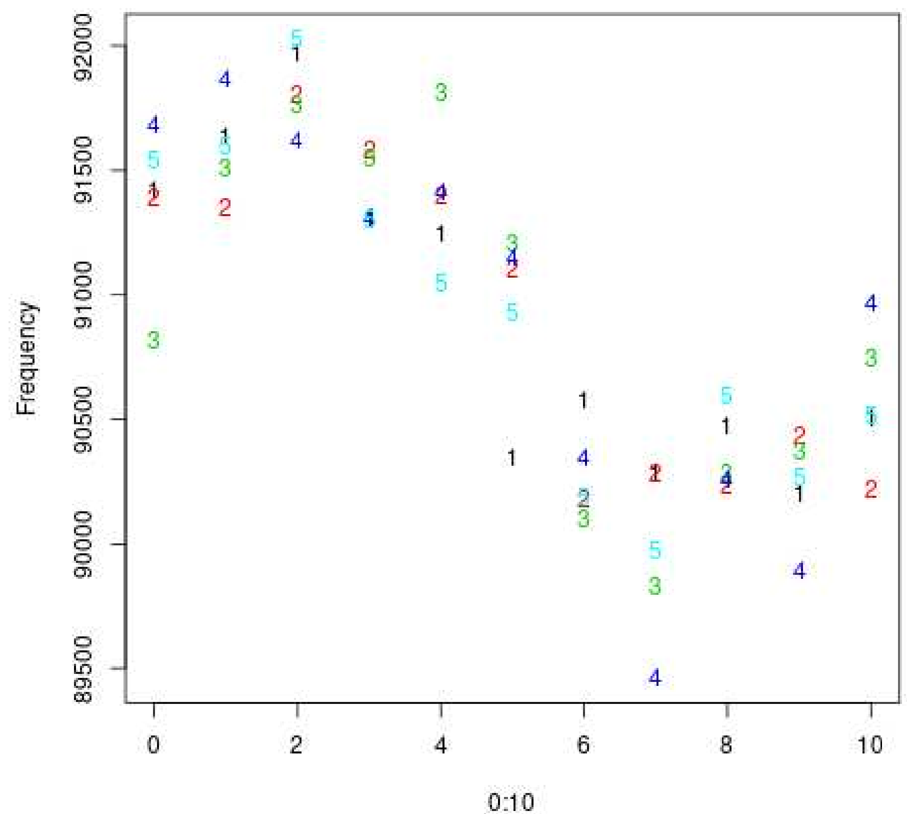




Hodím 10x kostkou, hodnoty sečtu modulo 11.







1. hodim kostkou $1\text{-}3 \to m = 0$

   $4\text{-}6 \to m = 5$

2. hodím kostkou $c = \text{hod} - 1$, pokud padne 6, házím znovu

celková hodnota: $m + c$

0	1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	41	42	43	44	45
21	22	23	24	25	26	51	52	53	54	55
31	32	33	34	35	36	61	62	63	64	65

- Musíme použít minimálně dva hody, neboť hod kostkou má $|S| = 6$, zatímco generátor má $|S| = 11$, pokud použijeme dva hody kostkou a budeme rozlišovat hod první a druhou kostkou, máme počet stavů $|S|^2 = 36$.
- Nyní je potřeba najít dobré zobrazení z jednoho prostoru. Očíslujme si jednotlivé výsledky dvou hodů kostkou ((1,1);(1,2);…;(6,6) → 1;2;…;36), a 33 stavů použijeme pro výsledky generátoru (například přes %11), tři zbývající budou neplatné a v takovém případě znovu dvakrát hodíme kostkou. Tím problém vyřešíme, ovšem s pravděpodobností 1/12 budeme házet znovu a teoreticky se nemusíme zastavit, ovšem v průmerném případě zastavíme ve E(počet dvojhodů) = 1 + 1/12.

Př. 10/11b: největší společný dělitel

Vypočtěte největší společný dělitel
a) $GCD(220, 284)$,
b) $GCD(\binom{30}{10}, \binom{31}{9})$,
c) $GCD(2^{100}, 100!)$.

b) $30045015 = 1 \cdot 20160075 + 9\,884\,940$

$20\,160\,075 = 2 \cdot 9\,884\,940 + 390\,195$

$9\,884\,940 = 25 \cdot 390\,195 + 130\,065$

$390\,195 = 3 \cdot \underline{130\,065} + 0$

$$\binom{30}{10} = \frac{30!}{10!\cdot 20!}$$

$$\binom{31}{9} = \frac{31!}{9!\cdot 22!} \overset{\cdot \frac{10}{10}}{=\!=} \frac{10\cdot 31!}{10\cdot 9!\cdot 22!} = \frac{31\cdot 10\cdot 30!}{10!\cdot 22\cdot 21\cdot 20!} = \frac{31\cdot 10}{22\cdot 21}\cdot \overbrace{\frac{30!}{10!\cdot 20!}}^{\binom{30}{10}\ \parallel}$$

Největším společným dělitelem je komb. číslo $\binom{30}{10}$.

Př. 10/11c: největší společný dělitel

Vypočtěte největší společný dělitel

a) $GCD(220, 284)$,

b) $GCD(\binom{30}{10}, \binom{31}{9})$,

c) $GCD(2^{100}, 100!)$.

c) $2^{100} = 2^{100}$

Legendre's formula: $v_2(100!) = \sum_{i=1}^{\infty} \left\lfloor \frac{100}{2^i} \right\rfloor =$

$$= \left\lfloor \frac{100}{2} \right\rfloor + \left\lfloor \frac{100}{4} \right\rfloor + \left\lfloor \frac{100}{8} \right\rfloor + \left\lfloor \frac{100}{16} \right\rfloor + \left\lfloor \frac{100}{32} \right\rfloor + \left\lfloor \frac{100}{64} \right\rfloor = 50 + 25 + 12 + 6 + 3 + 1 = 97$$

Př. 10/14: modulární umocňování - kód

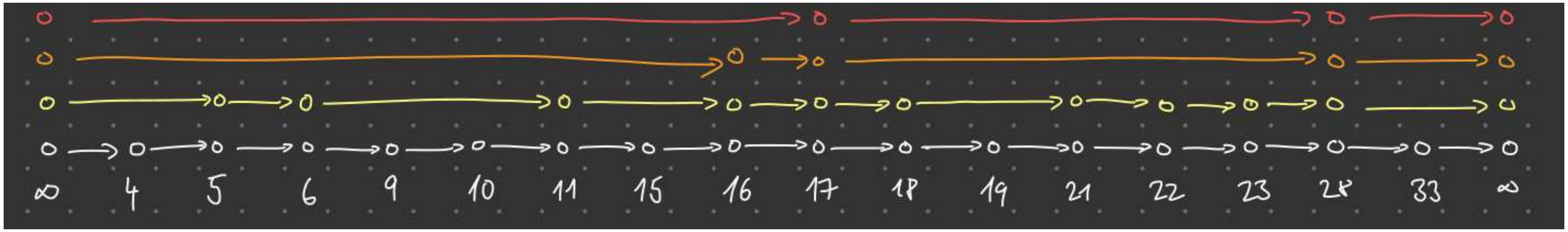
Uvedený kód počítá celočíselnou mocninu x^n. Popište, jak jej upravíte, aby počítal $x^n \ mod \ m$, pro kladné celé m. Minimalizujte riziko přetečení.

```
 BinPower(int x, int n) {
  int r = 1, y = x;
  while (n > 1) {
    if (n % 2 == 1) r *= y;
    y *= y;
    n /= 2;
  }
 return r*y;
}
```

Skip list. B-stromy.

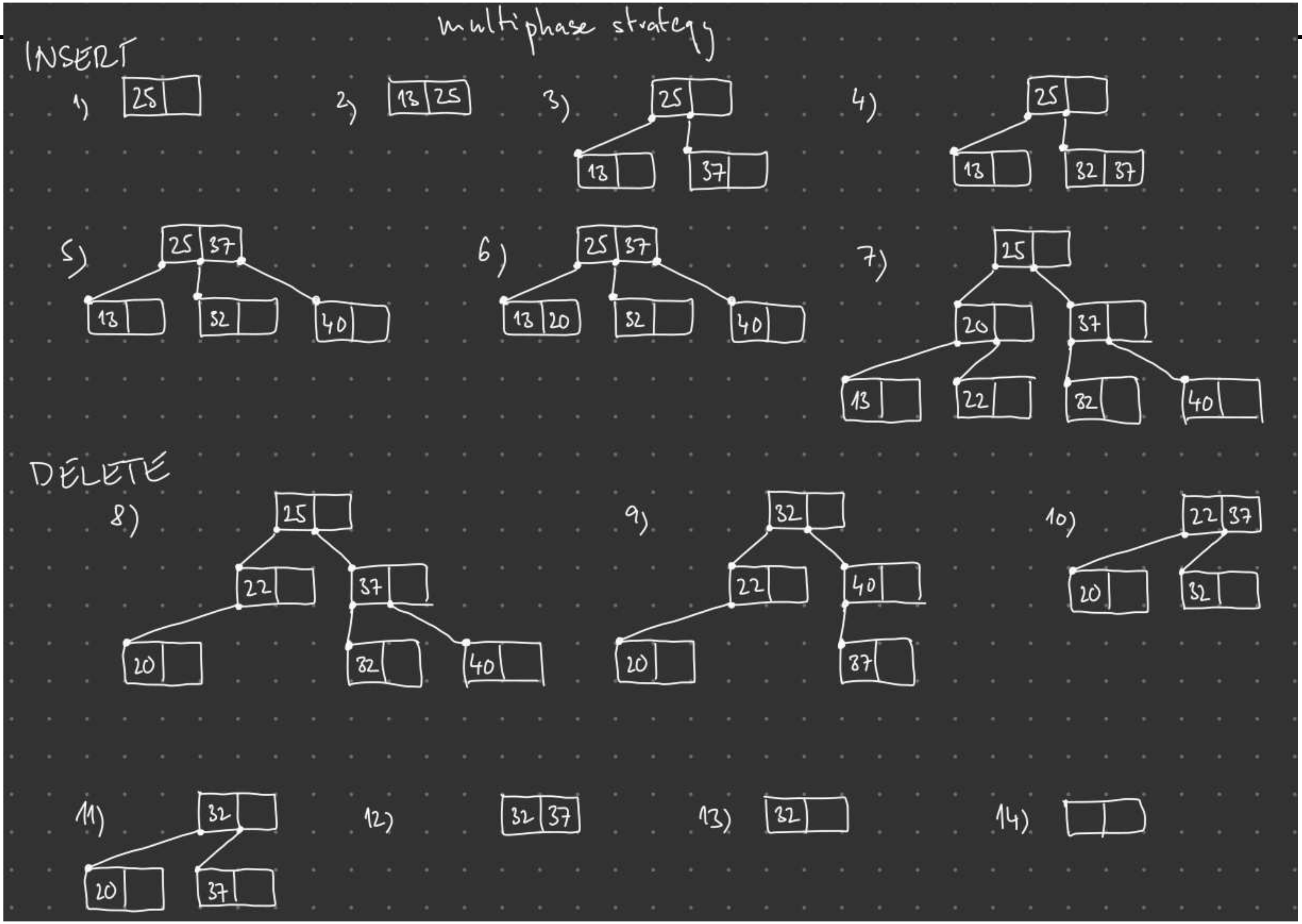
Př. 11/1: skip list - konstrukce

Sestavte skip list, který je nejprve prázdný a dále do něj vkládáte dané klíče v uvedeném pořadí. Číslo za klíčem v závorce uvádí úroveň (level) klíče, tj. kolikrát byla hozena mince, než padl rub (včetně rubu): 16(3), 23(2), 18(2), 5(2), 15(1), 19(1), 33(1), 11(2), 21(2), 4(1), 22(2), 6(2), 17(4), 10(1), 9(1), 28(4).



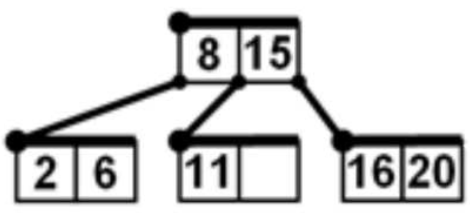
Př. 11/14: konstrukce a destrukce B stromu

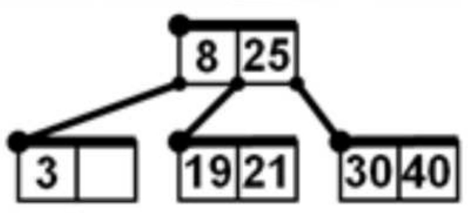
B-strom je řádu k, pokud každý jeho uzel, kromě kořene, musí obsahovat alespoň k klíčů a zároveň může obsahovat nejvýše $2k$ klíčů. Vybudujte B-strom řádu 1 tak, že do prázdného stromu vložíte v uvedeném pořadí klíče $25, 13, 37, 32, 40, 20, 22$. Dále tento strom zrušte, a to tak, že jednotlivé klíče klíče odstraníte v pořadí $13, 25, 40, 22, 20, 37, 32$. Nakreslete strom po každé operaci Insert a Delete.


multiphase strategy
INSERT
1)
25
2)
13 25
3)
25
13
37
4)
25
13
32 37
5)
25 37
13
32
40
6)
25 37
13 20
32
40
7)
25
20
37
13
22
32
40
DELETE
8)
25
22
37
20
32
40
9)
32
22
40
20
37
10)
22 37
20
32
11)
32
20
37
12)
32 37
13)
32
14)

Př. 11/11: B stromy

Do B-stromu znázorněného na levém resp. pravém obrázku vložíme postupně klíče 14, 10, resp. 7, 5. Jaké klíče pak bude obsahovat kořen stromu?





Př. 11/12: izomorfní B stromy

Dva prázdné B-stromy řádu 1 (max. 2 klíče v uzlu) jsou izomorfní. Neprázdný B-strom B1 řádu 1 s kořenem K1 je izomorfní s neprázdným B-stromem B2 řádu 1 s kořenem K2 právě tehdy, když zároveň platí 1. a 2.:

1. K1 obsahuje stejný počet klíčů jako K2
2. Levý podstrom K1 je izomorfní s levým podstromem K2, pravý podstrom K1 je izomorfní s pravým podstromem K2 a prostřední podstrom K1, pokud existuje, je izomorfní s prostředním podstromem K2.

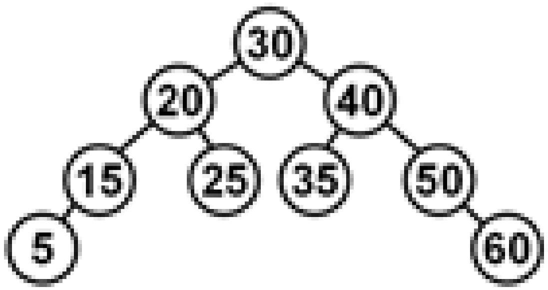
Určete počet navzájem neizomorfních B-stromů řádu 1 s A) 0, B) 1, C) 3, D) 4, E) 7 uzly.

Př. 11/16: plněné B stromy

Je dán a) B-strom, b) B+ strom. Strom je řádu 10 a máme do něj umístit 100 000 klíčů. Jaký je maximální a minimální možný počet uzlů tohoto stromu? Jaká je maximální a minimální možná hloubka tohoto stromu?

Př. 12/1: AVL stromy

Rozhodněte, zda a jaká rotace bude použita během operací DELETE klíčů 5, 25, 35 (v tomto pořadí) nad AVL stromem:



Př. 12/2: Splay stromy

Do nejprve prázdného stromu splay tree vkládejte postupně klíče 2, 7, 1, 4, 3, 9, 5, 6. Nakreslete strom po každém vložení.

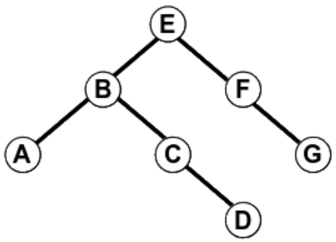
Př. 12/3: Splay stromy

Splay tree obsahuje $2^n - 1$ klíčů s hodnotou $1, 2, 3, \ldots 2^n - 1$ a je ideálně vyvážený, to jest má hloubku $n - 1$. Po vyhledání prvku s klíčem 1 se tento prvek přesune do kořene stromu. Jakou hloubku bude mít výsledný strom? Řešte zvlášť pro sudé a liché n.

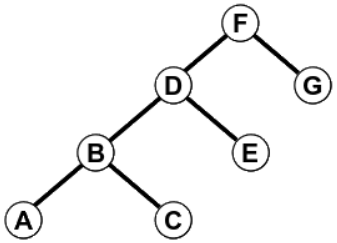
Př. 12/4: Červenočerné stromy

Navrhněte červenočerné obarvení daných stromů tak, aby vznikl korektní červenočerný strom. Prázdné (nil) listy nejsou zobrazeny.

A.



B.



C.