Informatika – navazující magisterské studium
Přijímací zkouška z informatiky – 2015 – varianta A

Každá úloha je hodnocena maximálně 25 body.
Všechny své odpovědi zdůvodněte!

1. Určete počet různých binárních vyhledávacích stromů s přesně 6 uzly, které obsahují hodnoty 27, 1, 3452, 815, 29 a 100.

2. Převeďte do disjunktivní normální formy následující logickou formuli:

$$\sim((y \,\&\, \sim(z \,\&\, u)) \vee (z \,\&\, (y \vee \sim x)))$$

Znaky x, y, z, u označují logické proměnné, znak & představuje logickou spojku konjunkce, znak ∨ označuje disjunkci, symbolem ~ x značíme negaci proměnné x.

a) Nalezněte jakékoliv řešení.
b) Nalezněte řešení ve tvaru disjunkce tvořené nejvýše dvěma členy.

3. Navrhněte deterministický konečný automat nad abecedou {0, 1}, který přijímá všechna taková slova, která začínají dvojicí stejných znaků a zároveň končí dvojicí stejných znaků (znaky na konci přijímaného slova mohou být jiné, než znaky na začátku tohoto slova). Například slova 00, 111, 00100, 11010010000 automat přijme, zatímco slova 1, 0101, 110110 nepřijme. Přechodovou funkci automatu zapište ve tvaru tabulky a automat znázorněte ve tvaru přechodového diagramu. Navrhněte co nejjednodušší automat, tzn. takový, který bude mít minimální počet stavů.

4. Je dán následující program (obě zadání v Pascalu a v jazyce C jsou ekvivalentní):

```
program A;
var P: array[0..200] of integer;
    i: integer;
begin
  for i:=0 to 200 do P[i] := i;
  for i:=1 to 200 do P[i] := P[i] + P[i-1];
  for i:=1 to 200 do write(P[i], ' ');
end.
```

```
main() /* A */
{
  int p[201];
  int i;
  for(i = 0; i <= 200; i++) p[i] = i;
  for(i = 1; i <= 200; i++) p[i] = p[i] + p[i-1];
  for(i = 1; i <= 200; i++) printf("%d ", p[i]);
}
```

a) Určete, na kterém indexu v poli P bude po skončení výpočtu hodnota 200.
b) Určete, na kterém indexu v poli P bude po skončení výpočtu hodnota 5050.
c) Určete, na kterém indexu v poli P bude po skončení výpočtu hodnota 22155.
Nalezněte vždy všechna řešení a zdůvodněte, proč žádné jiné řešení neexistuje.

Řešení přijímací zkoušky z informatiky – 2015 – varianta A

1. Stavíme binární vyhledávací stromy s N=6 různými hodnotami, na konkrétních hodnotách uložených v uzlech vůbec nezáleží. Počty všech různých BVS s N hodnotami odvodíme dynamickým programováním podle rostoucí hodnoty N. Pro N=0 existuje zjevně jediný BVS, pro N=1 také jeden, pro N=2 jsou stromy dva. Pro každé další N postupujeme takto: V kořeni BVS může být libovolná z uvažovaných N hodnot, levý a pravý podstrom jsou potom BVS s K uzly a s N-K-1 uzly (kde K závisí na volbě hodnoty v kořeni). Počty takových podstromů již známe, stačí tedy vynásobit počet možností pro levý a pro pravý podstrom a tyto dílčí součiny posčítat pro všechna K od 0 do N-1. Postupně vychází, že pro N=3 existuje 5 různých BVS, pro N=4 jich je 14, pro N=5 existuje 42 BVS a konečně pro N=6 dostáváme **132** různých BVS.

2. Zadanou logickou formuli upravíme, použijeme přitom de Morganova pravidla:

~((y & ~(z & u)) ∨ (z & (y ∨ ~x)))

~ (y & ~(z & u)) & ~(z & (y ∨ ~x)))

(~y ∨ (z & u)) & (~z ∨~(y ∨ ~x))) atd.

Postupnými úpravami dostaneme výsledný tvar

(x & ~y) ∨ (~y & ~z)

Můžeme také postupovat pomocí tabulky logických hodnot, do níž vypíšeme všech 16 možností pro hodnoty logických proměnných x, y, z, u. Formuli v disjunktivní normální formě pak získáme jako disjunkci těch sloupců tabulky, v nichž nabývá zadaná formule pravdivostní hodnoty 1. Rychle a přímočaře totéž řešení získáme pomocí Karnaughovy mapy.

3. Počáteční stav A není koncový, automat přijímá slova délky aspoň 2. Další dva stavy B, C rozlišují první přečtený znak vstupního slova. Pokud na vstupu následuje odlišný znak, automat přejde do nekoncového stavu D, ve kterém již setrvá až do přečtení celého slova. Naopak po přečtení druhé nuly přejde automat ze stavu B do koncového stavu E, ve kterém čte i případné další nuly. Obdobně po přečtení druhé jedničky přejde automat ze stavu C do koncového stavu F, ve kterém čte i případné další jedničky. Následně automat předchází mezi stavy E, F, G, H, které určují poslední přečtenou dvojici znaků. Minimalitu počtu stavů ověříme redukcí sestrojeného konečného automatu.

	0	1	
→ A	B	C	počáteční stav
B	E	D	jedna 0 na začátku slova
C	D	F	jedna 1 na začátku slova
D	D	D	slovo začíná dvěma různými znaky
← E	E	H	slovo zatím končí 00
← F	G	F	slovo zatím končí 11
G	E	H	slovo zatím končí 10
H	G	F	slovo zatím končí 01

4. Po provedení prvního cyklu platí P[i] = i pro všechna i od 0 do 200. Ve druhém cyklu se počítají kumulativní sumy z hodnot prvků pole P, takže po jeho skončení bude P[i] rovno součtu všech čísel od 0 do i, což je i.(i+1)/2. Uvedená funkce je rostoucí, takže všechny hodnoty v poli P budou uspořádány vzestupně.

a) Hledáme, pro které i platí i.(i+1)/2 = 200, neboli i.(i+1) = 400. Pro i=19 platí i.(i+1) = 19.20 = 380, pro i=20 platí i.(i+1) = 20.21 = 420. Vzhledem k rostoucímu uspořádání hodnot v poli P proto nemůže existovat žádné i splňující uvedenou rovnost. Hodnota 200 v poli P **vůbec nebude**.

b) Postupuje obdobně a dostáváme rovnost i.(i+1) = 10100. Hodnotu 10100 můžeme zjevně rozložit na součin 100.101, takže vyhovuje i = **100**. Vzhledem k rostoucímu uspořádání prvků v poli P je to jediné řešení. Úlohu lze řešit také nalezením kořenů uvedené kvadratické rovnice, pouze kořen 100 je kladný.

c) Pro nejvyšší index v poli i=200 platí P[i] = i.(i+1)/2 = 200.201/2 = 20100. Prvek P[200] je nejvyšší hodnotou v poli P, takže hodnota 22155 v poli P **vůbec není**. (V případě většího pole P by hodnota 22155 byla umístěna na indexu 210.)