Informatika – navazující magisterské studium
Přijímací zkouška z informatiky – 2022 – varianta A

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

1. Určete počet různých binárních vyhledávacích stromů, které jsou tvořeny šesti vrcholy s hodnotami 11, 22, 33, 44, 55, 66.

2. Navrhněte deterministický konečný automat nad abecedou {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, který přijímá všechna slova představující dekadický zápis celého čísla beze zbytku dělitelného čtyřmi. V zápisu čísla jsou povoleny vedoucí nuly. Automat tedy přijme například slova 8, 0032, 1112, zatímco slova 2, 015, 186 nepřijme. Automat nepřijme ani prázdné slovo. 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.

3. Prvočíselným rozkladem kladného celého čísla *N* rozumíme rozklad čísla *N* na součin prvočísel. Chceme určit počet různých prvočísel obsažených v prvočíselném rozkladu zadaného čísla *N* (kde *N* > 1). Popište slovně co nejefektivnější algoritmus řešení a zapište ho ve tvaru funkce v programovacím jazyce Pascal, C/C++, Java, C# nebo Python. Hodnotí se nejen správnost, ale také časová efektivita zvoleného řešení.

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

```
program AA;
var i, j, s: integer;
begin
  read(n);
  s := 0;
  for i := 1 to n do
    for j := i to 3*i do
      s := s + 2*i-j+1;
  write(s)
end.
```

```
void main(void)
{
  int i, j, s;
  scanf("%i", &n);
  s = 0;
  for(i = 1; i<=n; i++)
    for(j = i; j<=3*i; j++)
      s += 2*i-j+1;
  printf("%i", s);
}
```

```
n = int(input())
s = 0
for i in range(1, n+1):
    for j in range(i, 3*i+1):
        s += 2 * i - j + 1
print(s)
```

a) Určete, jaký výsledek obdržíme při výpočtu se vstupní hodnotou $n = 98$.
b) Určete všechny takové vstupní hodnoty *n*, pro které výpočet programu skončí s výsledkem 120.
c) Vyjádřete co nejjednodušším způsobem závislost výsledné hodnoty *s* na vstupní hodnotě *n*.

Řešení přijímací zkoušky z informatiky – 2022 – varianta A

1. Počet různých binárních vyhledávacích stromů (dále BVS) s n vrcholy označíme p_n. Jistě platí $p_0 = 1$, $p_1 = 1$. Odvodíme obecný vztah pro p_n, z něhož pak přímo získáme požadovaný výsledek p_6.

Každý BVS má kořen, jeho levým podstromem je BVS s i vrcholy (hodnoty menší než v kořenu) a pravým podstromem je BVS s n-i-1 vrcholy (hodnoty větší než v kořenu) pro nějaké i od 0 do n-1. Pro každou volbu kořene tedy existuje $p_i.p_{n-i-1}$ různých BVS, kde i je počet hodnot menších než hodnota umístěná do kořene. Celkový počet všech BVS s n vrcholy je proto součtem všech takových $p_i.p_{n-i-1}$ pro i od 0 do n-1.

Podle uvedeného vztahu budeme postupně počítat p_2, p_3, p_4, p_5, p_6:

n	0	1	2	3	4	5	6
p_n	1	1	2	5	14	42	**132**

2. Číslo je dělitelné čtyřmi, právě když je čtyřmi dělitelné jeho poslední dvojčíslí. To znamená, že čísla dělitelná čtyřmi končí buď některou z cifer 2, 6 následující po liché cifře, nebo některou z cifer 0, 4, 8 následující po sudé cifře (příp. 0, 4, 8 může být jedinou cifrou čísla). V automatu proto musíme rozlišovat tři skupiny cifer: {1,3,5,7,9}, {2,6}, {0,4,8}. Cifry z téže skupiny se z hlediska přechodové funkce chovají stejným způsobem. Pomocí stavů automatu rozlišíme, jakými ciframi končí již zpracovaná část vstupního slova:

stav A – poslední cifra je lichá

stav B – poslední cifra je sudá; je z {2,6} a následuje po liché cifře,
nebo je z {0,4,8}a následuje po sudé cifře (příp. je to jediná cifra čísla)

stav C – poslední cifra je sudá; je z {0,4,8} a následuje po liché cifře,
nebo je z {2,6} a následuje po sudé cifře (příp. je to jediná cifra čísla)

Podle definice dělitelnosti čtyřmi je právě stav B koncový. Stav C nám poslouží zároveň jako počáteční stav automatu. Automat tedy bude mít pouze tři stavy. Minimalitu počtu stavů snadno ověříme redukcí sestrojeného konečného automatu.

	1,3,5,7,9	**2,6**	**0,4,8**
A	A	B	C
← **B**	A	C	B
→ C	A	C	B

3. Základní algoritmus: Zavedeme si pomocnou proměnnou D=2. Dokud bude hodnota N dělitelná číslem D, budeme N opakovaně dělit číslem D. Když už N nebude dělitelné, posuneme se k dalšímu potenciálnímu děliteli zvýšením D o 1. Průběžně si počítáme, kolika různými hodnotami D jsme N vydělili. Popsaný proces ukončíme ve chvíli, když N=1.

Možnosti zrychlení:

a) Nejprve zvlášť ověříme dělitelnost N dvěma a poté v cyklu zkoušíme pouze liché dělitele D od 3 výše (zvyšováním hodnoty D o 2).

b) Popsaný proces ukončíme již ve chvíli, když aktuální hodnota D převýší aktuální hodnotu druhé odmocniny z N. Zbývající hodnota uložená v proměnné N je pak posledním prvočinitelem.

Příklad zápisu algoritmu ve tvaru funkce:

```
def pocet_prvocinitelu(n):
    pocet = 0
    d = 2
    while n > 1:
        if n % d == 0:
            pocet += 1
            while n % d == 0: n //= d
        else:
            d += 1
            if d * d > n:
                pocet += 1
                break
    return pocet
```

4. Nejprve vyřešíme obecný vztah požadovaný v části c). Pro každou hodnotu i se proměnná s změní takto:

- přičte se k ní hodnota $2i+1$ tolikrát, kolik se provede iterací vnitřního cyklu, což je $2i+1$, celkem se tedy s zvýší o $(2i+1)^2$

- odečte se od ní součet všech j z rozmezí od i do $3i$, což je podle vzorce pro součet aritmetické posloupnosti $(i+3i)*(2i+1)/2 = 2i(2i+1)$

Proměnná s se tedy pro každé i zvýší vždy o $(2i+1)^2 - 2i(2i+1) = 2i+1$

Sečtením výrazů $2i+1$ pro všechna i od 1 do n dostaneme výslednou hodnotu proměnné s:

$\sum(2i+1) = 2\sum i + \sum 1 = n(n+1) + n = n(n+2)$

Odvozený výsledný vztah $\boldsymbol{s = n(n+2)}$ platí ovšem pouze pro nezáporné vstupní hodnoty n, zatímco pro záporná n se cyklus neprovede ani jednou a výsledkem výpočtu je $\boldsymbol{s = 0}$.

a) Úlohu vyřešíme přímým dosazením vstupní hodnoty 98 do odvozeného vzorce, dostaneme výsledek s = **9800**.

b) Výsledná hodnota 120 je kladná, takže také hledaná vstupní hodnota n musí být kladná. Funkce $n(n+2)$ je pro kladná n rostoucí, proto může vyhovovat nejvýše jedna taková vstupní hodnota. Pokud opravdu existuje, získáme ji jako kladný celý kořen kvadratické rovnice $n(n+2) = 120$. Na první pohled je zřejmý rozklad 10.12=120, případně můžeme rovnici vyřešit pomocí diskriminantu. V obou případech dostaneme kladný celý kořen 10 (a druhý záporný kořen -12, který podmínkám úlohy nevyhovuje). Výslednou hodnotu s = 120 tedy získáme pouze pro vstupní hodnotu **10**.