A Jméno _____

1. (2 body, více správných odpovědí, 1 chyba = 1 bod, více chyb = 0 bodů)

V jaké třídě složitosti je funkce `push` v následujícím programu vzhledem k velikosti pole `p` (velikost budeme značit n)? Předpokládejte, že funkce `new` pracuje v konstantním čase vzhledem k velikosti pole `p`. Dále předpokládejte, že velikost hodnoty proměnné `N` vždy před a po provedení funkce `push` koresponduje s velikosti pole `p`. Proměnná `i` je vždy v intervalu `-1` až `N`.

```
public class MyHeap {

      int N = 0;
      int i = -1;
      int [] p;

      public void push (int key)
      {
            if (++i >= N) {
                  int m = N*2+1;
                  int [] t = new int[m];
                  for(int j = 0; j<N; j++, t[j] = p[j]);
                  N = m;
                  p = t;
            }
            p[i] = key;
            int j = i;
            while (p[j >> 1] > p[j]) {
              key = p[j];
              p[j] = p[j >> 1]
              p[j >>= 1] = key;
          }
   }
}
```

a) $O(1)$
b) $O(n*\log(n))$
c) $O(n^2)$
d) $\Omega(1)$
e) $\Omega(\log(n))$
f) $\Omega(n^2)$
g) $\Theta(n^2)$
h) $\Theta(n)$
i) $\Theta(1)$

2. (2 body, jedna správná odpověď)

Funkce:

```
int foo(int x, int y) {
 if (y>0) return foo(x, y-1)+x;
 return 0;
}
```

a) pro kladná y vrátí y, jinak vrátí x.
b) spočte rozdíl x–y, je-li y nezáporné.
c) spočte rozdíl y–x, je-li y nezáporné.
d) vrátí hodnotu svého většího parametru.
e) vynásobí x a y, je-li y nezáporné.

3. (2 body, jedna správná odpověď)
Funkce $H(n,m)$ je definována níže. Vypočtěte ručně hodnotu $H(2,2)$.

$$H(n,m) = \begin{cases} m+1 & \text{pro } n = 0, \\ H(n-1,1) & \text{pro } n > 0 \text{ a } m = 0, \\ H(n-1,H(n,m-1)-1) & \text{pro } n > 0 \text{ a } m > 0. \end{cases}$$

a) 0
b) 1
c) 2
d) 3
e) 4
f) 5
g) hodnota není definovaná

4. (2 body, více správných odpovědí, 1 chyba = 1 bod, více chyb = 0 bodů)
Určete, do jaké třídy složitosti náleží následující rekurentně definovaná funkce T:

$$T(n) = 4T\left(\frac{n}{2}\right) + n^3$$

a) $T(n) \in O(n^2)$
b) $T(n) \in O(n \cdot \log(n))$
c) $T(n) \in O(n^3)$
d) $T(n) \in O(n^n)$
e) $T(n) \in \Omega(n^2)$
f) $T(n) \in \Omega(n^3)$
g) $T(n) \in \Omega(n)$
h) $T(n) \in \Omega(n \cdot \log(n))$
i) $T(n) \in \Theta(n^2)$
j) $T(n) \in \Theta(n \cdot \log(n))$
k) $T(n) \in \Theta(n^2 \cdot \log(n))$
l) $T(n) \in \Theta(n^3)$

5. (1 bod, jedna správná odpověď)
Kolize u hashovací (nebo také rozptylovací) funkce $h(k)$:
a) je situace, kdy pro dva stejné klíče k vrátí $h(k)$ různou hodnotu
b) je situace, kdy funkce $h(k)$ při výpočtu havaruje (skončí chybou nebo výjimkou)
c) je situace, kdy v otevřeném hashování (open-address hashing) dojde dynamická paměť
d) je situace, kdy pro dva různé klíče k vrátí $h(k)$ stejnou hodnotu

6. (1 bod, jedna správná odpověď)
Metoda perfektního hashování (perfect hashing):
a) dokáže uložit libovolný předem neznámý počet klíčů
b) ukládá prvky s klíči v dynamické paměti
c) vždy ukládá prvky do lineárního spojového seznamu
d) nemá problém s kolizemi, protože nevznikají

7. (1 bod, jedna správná odpověď)
Metoda otevřeného hashování s dvojitým hashováním (double hashing):
a) je metoda zmenšující délku clusterů u otevřeného hashování
b) má stejnou pravděpodobnost vzniku dlouhých clusterů jako metoda otevřeného hashování s lineárním přidáváním (linear probing)
c) je metoda ukládání klíčů na dvě různá místa
d) má vyšší pravděpodobnost vzniku dlouhých clusterů než metoda otevřeného hashování s lineárním přidáváním (linear probing)

8. (1 bod, jedna správná odpověď)

Metoda externího hashování (extendable hashing)

a) dokáže uložit pouze předem známý počet klíčů
b) ukládá synonyma do samostatných seznamů v dynamické paměti
c) ukládá synonyma spolu s ostatními klíči v poli
d) dokáže uložit libovolný předem neznámý počet klíčů

9. (2 body, jedna správná odpověď)

Jak vypadá hashovací tabulka po vložení následujících klíčů A, B, C, D, E, F, G (v tomto pořadí) metodou LISCH (late insert standard coalesced hashing), když známe hodnoty hashovací funkce *h* (viz níže) a velikost celé hashovací tabulky (M'=10)?

h(A)=1, h(B)=3, h(C)=1, h(D)=4, h(E)=3, h(F)=10, h(G)=1

a)

1	A	7
2		
3	B	9
4	D	
5		
6		
7	G	10
8	F	
9	E	
10	C	8

b)

1	G	7
2		
3	E	9
4	D	
5		
6		
7	C	10
8	A	
9	B	
10	F	8

c)

1	A	7
2		
3	B	9
4	D	
5		
6		
7	G	8
8	F	10
9	E	
10	C	

d)

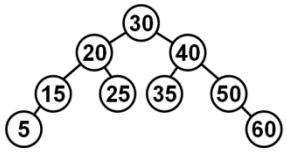
1	A	10
2		
3	B	9
4	D	
5		
6		
7	G	
8	F	7
9	E	
10	C	8

10. (1 bod, jedna správná odpověď)

Dynamické programování je metoda, která

a) umožňuje řešit všechny problémy jako metoda rozděl a panuj, ale s lepší nebo stejnou asymptotickou časovou složitostí
b) umožňuje řešit problémy pomocí průběžné dynamické alokace řešení podproblémů v paměti
c) umožňuje řešit pouze úlohy s různě velkými vstupy
d) umožňuje efektivně řešit problémy pomocí rozkladu na podproblémy, které jsou nezávislé (nesdílejí společný podprostor řešení).

11. (2 body, jedna správná odpověď)
Na obrázku je uveden BVS, který v průběhu práce vyvažujeme (AVL strom). Ten nyní upravíme tak, že z něj odstraníme operací Delete uzly s klíči 5, 25, 35 v tomto pořadí. Rozhodněte, zda a jaká rotace bude během této úpravy použita:



a) L rotace
b) R rotace
c) LR rotace
d) RL rotace
e) žádná rotace

12. (2 body, jedna správná odpověď)
Binární strom (nikoli nutně pravidelný) má přesně 2^k vnitřních uzlů a 2^k listů. Hloubka tohoto stromu leží právě v intervalu (připomínáme, že kořen má hloubku 0)

a) $< \log_2 k, k >$
b) $< \log_2 k, k+1 >$
c) $<k, 2^k+1>$
d) $<k+1, 2^k>$
e) $< 2^k, 2^{k+1}>$

13. (1 bod, jedna správná odpověď)
Pole délky n obsahuje hodnoty 1, n, $n-1$, $n-2$, , $n-3$, ... 4, 3, 2 v uvedeném pořadí. Počet porovnání dvojic čísel, které provede Insert sort při řazení tohoto pole do neklesajícího pořadí, je

a) $\log_2(n)$
b) $n - 1$
c) $2(n^2 + n)$
d) $(n^2 - n)/2$
e) $n^2/2$

14. (1 bod, jedna správná odpověď)
Paměťové nároky Merge sort-u při řazení pole délky n jsou uvedeny níže. Předpokládáme, že jeden prvek pole obsazuje jednotkovou velikost paměti

a) konstantní, nezávislé na n
b) $\log(n)$ + konstanta
c) n + konstanta
d) $c \cdot n$ + konstanta, $c > 1$
e) $(n^2 - n)/2$ + konstanta

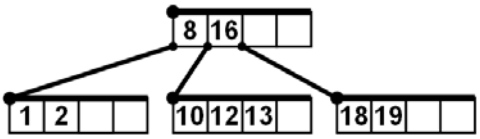
15. (2 body, jedna správná odpověď)
Na vstupu Heap sortu je neseřazené pole uvedené níže.
28 23 29 15 30 13 21 14 19
V první fázi řazení Heap sort vytvoří haldu s nejmenším prvkem na začátku pole. Tato halda je

a) 13 14 15 28 21 30 29 19 23
b) 13 14 15 21 30 29 28 23 19
c) 13 14 21 28 30 29 23 15 19
d) 13 14 21 15 30 29 28 23 19
e) 13 21 14 15 30 29 28 19 23

16. (1 bod, jedna správná odpověď)
Vstupní pole pro řazení obsahuje hodnoty v uvedeném pořadí:
12 9 11 9 8 8 8 9 8 12
Pole řadíme pomocí Counting sortu. Toto řazení používá pomocné pole, které se nejprve plní četnostmi vstupních hodnot a pak se ještě dále zpracovává. Těsně předtím, než se začne plnit výstupní pole (které je indexováno od 0), je obsah pomocného pole následující

a) od indexu 0 do indexu 9: 8 8 8 8 9 9 9 11 12 12
b) od indexu 4 do indexu 10: 12 12 9 9 9 11 8
c) od indexu 8 do indexu 12: 3 6 6 7 9
d) od indexu 8 do indexu 12: 2 1 0 3 4
e) od indexu 9 do indexu 14: 10 4 3 0 1 2

17. (2 body, jedna správná odpověď)
Z B-stromu znázorněného na obrázku odebereme postupně klíče 18, 2. Pak bude kořen obsahovat klíč/klíče



a) 8
b) 8, 16
c) 10
d) 10, 12
e) 13

18. (2 body, jedna správná odpověď)
Jednotlivé klíče binárního stromu vypíšeme nejprve v pořadí Inorder a potom v pořadí procházení do šířky. Získáme tak posloupnosti (E, G, A, B, F, C, D), a (G, E, F, B, D, A, C). Po vypsání klíčů v pořadí Preorder získáme posloupnost

a) (G, E, F, B, A, D, C)
b) (G, E, A, B, F, C, D)
c) (G, B, D, A, E, F, C)
d) (G, B, A, E, F, C, D)
e) (A, B, C, D, E, F, G)

19. (1 bod, jedna správná odpověď)
Quick sort řadí následující šestiprvkové pole čísel.
5 7 9 8 2 1
Jako pivotní hodnotu volí první v pořadí tj. nejlevější. Jak bude pole rozděleno na „malé“ a „velké“ hodnoty po jednom průchodu polem? (Lomítko naznačuje místo dělení.)

a) **1 2 9 8 / 7 5**
b) **1 2 / 9 8 7 5**
c) **9 8 7 / 1 2 5**
d) **5 7 1 / 9 8 2**
e) **2 1 7 / 8 9 5**

20. (1 bod, jedna správná odpověď)
Radix sort řadí pole řetězců {acbc, ccbc, adda, daab, bcca, accb, bada, dacc, bbac}.
Po prvních dvou průchodech algoritmu bude seznam odpovídající znaku “d” obsahovat právě řetězce v uvedeném pořadí

a) adda, bada, dacc
b) daab, dacc, adda
c) daab, dacc
d) adda, bada
e) seznam bude prázdný