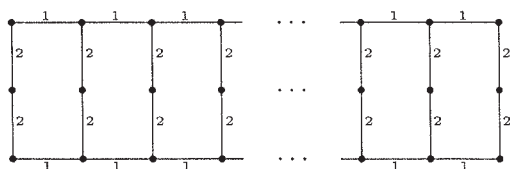


1

Černá výška červenočerného stromu je počet černých uzlů na každé jeho větvi.

- Nakreslete červenočerný strom s *minimálním možným* počtem uzlů, který má černou výšku 3. Ohodnoťte tento vyhledávací strom přirozenými čísly od jedné až do počtu uzlů.
- Nakreslete červenočerný strom s *maximálním možným* počtem uzlů, který má černou výšku 2. Ohodnoťte tento vyhledávací strom přirozenými čísly od jedné až do počtu uzlů.
- Přidejte do tohoto (většího) stromu uzel s číslem 0 a znázorněte jednotlivé kroky při vyvažování.

2



Graf na obrázku má $3k$ uzlů, $4k - 2$ hrany (kladné celé číslo k určuje počet svislých „příček“) a naznačené hranové ohodnocení. Popište všechny jeho *minimální* kostry. Kolik jich je? Vyjádřete jejich počet pomocí parametru k .

- 4

(b) Dále tento algoritmus modifikujeme tak, že začneme slučovat až od seřazených šestnácti prvků: Nejdrůbev posoupnost rozdělíme do $\frac{n}{16}$ úseků po 16 prvcích a každý z těchto úseků seřadíme pomocí řazení vkládáním, tj. algoritmem *Insert sort*. Potom pokračujeme původním algoritmem – seřazené šestnáctice slučujeme do dvaatřicetice, seřazené dvaatřicetice do čtyřiašedesátic atd., až bude seřazená celá posoupnost. Jaká bude časová složitost takto modifikovaného algoritmu? Zdůvodněte.

5

Máme je dáno kladné celé číslo n a čtvercovou matici A typu $n \cdot n$ obsahující pravdivostní hodnoty. Tato matice je maticí sousednosti orientovaného grafu s uzly $1, \dots, n$. K dispozici je jedno jednorozměrné přepisovatelné pole B délky n , do něhož ukládáme čísla uzlů. Toto pole má na počátku všechny prvky inicializované na „neuzel“ (reprezentovaný číslem 0). Napište proceduru `predecessors`, která bude mít jako parametr daný uzel k a která naplní pole B předchůdcovskými uzly stromu průchodu do hloubky ze zadaného uzlu k . Po skončení výpočtu `predecessors(k)` bude mít každý uzel i dosažitelný z uzlu k nastaveno B_i na číslo svého předchůdce v *DFS*-stromu, sám uzel k bude mít $B_k = k$. Pro uzly j nedosažitelné z k zůstane v B_j číslo 0.

6

Je dána maximová zleva zarovnaná binární halda uložená v poli A . Velikost haldy je uložena v globální proměnné n . Napište definici procedury `insertHeap`, která do této haldy přidá nový prvek p , a tedy zvětší haldu na $(n+1)$ -prvkovou. Procedura `insertHeap` musí mít co nejlepší časovou složitost. Určete a zdůvodněte tuto složitost.

```
n : nat;
A : array 1.. of real;
procedure insertHeap (p:real);
{
    :
}
```