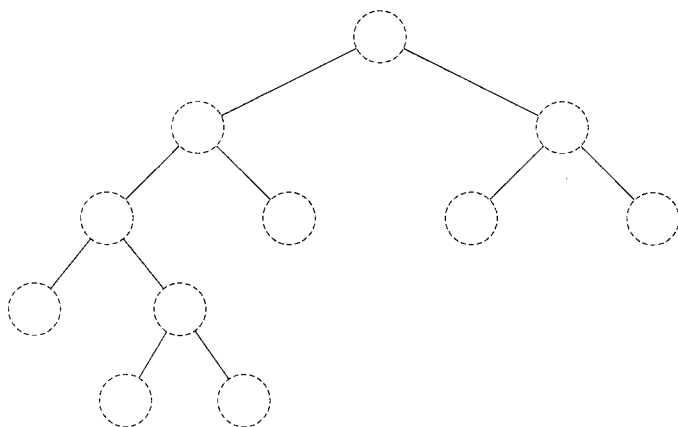
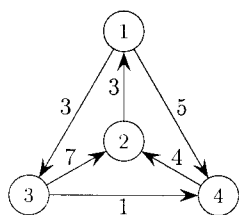


1



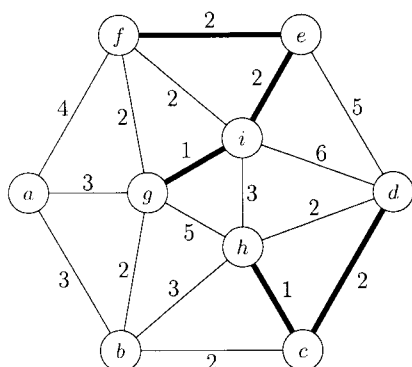
- (a) Do uzlů jedenáctiuzlového binárního stromu na obrázku vepište čísla 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, aby byl takto ohodnocený strom binárním vyhledávacím stromem (pozor, číslo 4 zde zatím není).
- (b) Kolika způsoby lze obarvit uzly tohoto stromu, aby vznikl červenočerný strom? Uvažte jeho černou hloubku.
- (c) Vyznačte jedno z těchto obarvení v obrázku. Černé uzly označte dvojitým kroužkem, červené uzly jednoduchým.
- (d) Nyní do stromu přidejte uzel s klíčem 4 a proveďte nutné úpravy stromu tak, aby zůstal červenočerným stromem. Kolik rotací uzlů je potřeba?
- (e) Popište tyto rotace a výsledný červenočerný strom nakreslete.

2



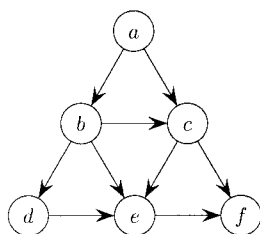
K orientovanému grafu na obrázku napište jeho matici vzdáleností D . Není-li uzel j z uzlu i dosažitelný, položte $D_{i,j} = +\infty$.

3



Neorientovaný graf s vyznačenými hranami vyjadřuje stav konstrukce minimální kostry podle Kruskalova algoritmu po přidání pěti hran. Vyznačené hrany budou patřit minimální kostře. V dalších třech krocích výpočtu budou přidány další tři hrany kostry, ale je více možností, kterými lze tyto tři hrany vybrat. Napište všechny tyto možnosti, tj. všechny takové trojice hran. Rozlišujte přitom i trojice vedoucí ke stejné minimální kostře, ale s různým pořadím vybíraných hran.

4



Orientovaný graf na obrázku procházíme do hloubky algoritmem *DFS* z jediného výchozího uzlu *a*. Napište všechna možná pořadí (uspořádané šestice začínající uzlem *a*), v nichž lze navštívit dosažitelné uzly. Pro každou takovou šestici uzlů nakreslete jí odpovídající strom předchůdců. Které šestice mají stejný strom?

5

Je dána minimová 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. Pomůcka: zdola

```
n : nat;
A : array 1.. of real;

procedure insertHeap (p:real);
{
    :
}
```

6

Máme k dispozici základní operace binárního stromu s číselně ohodnocenými uzly: `isEmpty` (predikát prázdnoty), `rootval` (hodnota kořene), `left`, `right` (levý a pravý podstrom). Napište definici funkce `isBST`, která otestuje, zda daný binární strom je vyhledávací. Nepředpokládejte žádnou konkrétní implementaci stromu, ale použijte jen uvedené metody.

Nápověda: `isBST(t)` = `bounds(t, $-\infty$, $+\infty$)`
`bounds(t, min, max)` = .. ~~.....~~