1.4.20 Nejkratší cesty z výchozího vrcholu r. Úloha: Najděte délky nejkratších cest z výchozího vrcholu r.

1.4.21 Obecné schema.

Vstup: orientovný graf $G = (V, E)$ a ohodnocení hran a.

Výstup: hodnoty $U(v)$ rovné $u(r, v)$.

1. (Inicializace.)
    $U(r) := 0$, $U(v) := \infty$ pro $v \neq r$;
2. (Zpracování hran.)
    Existuje-li hrana $e = (v, w)$ taková, že
        $U(w) > U(v) + a(e)$
            položíme $U(w) := U(v) + a(e)$.
3. (Ukončení.)
    Jestliže $U(w) \leq U(v) + a(e)$ pro každou hranu $e = (v, w)$
        stop.
    Jinak pokračuj krokem 2.

1.4.22 Tvrzení. Jestliže v grafu G neexistuje cyklus záporné délky a hodnota $U(v) \neq \infty$, pak $U(v)$ je délka některé cesty z vrcholu r do vrcholu v.

Nástin důkazu: Označme $U_t(y)$ hodnotu $U(y)$ v okamžiku t. Platí: jestliže v nějakém okamžiku t_k je $U_{t_k}(x) \leq \infty$, tak musí existovat sled

$$r = v_1, e_1, v_2, e_2, \ldots, v_{k-1}, e_{k-1}, v_k = x$$

a časové okamžiky $t_1 < t_2 < \ldots < t_k$ tak, že

$$U_{t_i}(v_i) = \sum_{j=1}^{i} a(e_j).$$

Nyní je třeba dokázat, že se nejedná o sled, ale o cestu. Kdyby se ve sledu opakoval vrchol, tj. kdyby např. $v_i = v_j$ pro $i < j$, pak $U_{t_i}(v_i) > U_{t_j}(v_j)$ a proto se dá dokázat, že $v_i, e_i, v_{i+1}, e_{i+1}, \ldots, v_j$ obsahuje cyklus záporné délky.

1.4.23 Věta. Jestliže graf G neobsahuje cyklus záporné délky a hodnoty $U(v)$ byly získány podle schematu 1.4.21, pak $U(v) = u(r, v)$.

Důkaz: Sporem. Kdyby tvrzení věty neplatilo, po skončení práce schematu by existoval vrchol v takový, že $U(v) > u(r, v)$. To také znamená, že $u(r, v) < \infty$. Vezměme nejkratší cestu C z vrcholu r do vrcholu v. Na této cestě je první vrchol výchozí a pro něj platí $U(r) = u(r, r)$, poslední vrchol je vrchol v, pro který $U(v) > u(r, v)$. Vezměme na cestě C první hranu $e = (x, y)$ takovou, že $U(x) = u(r, x)$ a $U(y) > u(r, y)$. Pro tyto dva vrcholy platí:

$$U(y) > u(r, y) = u(r, x) + a(x, y) = U(x) + a(x, y).$$

Tedy, obecné schema nemělo skončit, protože trojúhelníková nerovnost neplatí pro hranu $e = (x, y)$.

1.4.24 Nejkratší cesty mezi všemi dvojicemi vrcholů. Úkolem je najít celou matici vzdáleností (a ne jen jeden její řádek).

Označme množinu vrcholů grafu G $V = \{1, 2, \dots, n\}$. Floydův algoritmus (v literatuře též nazývaný Floyd-Warshallův algoritmus) je založen na konstrukci matic $\mathbf{U}_k = (u_k(i,j))$ řádu n pro $k = 0, 1, \dots, n$ s následující vlastnosti:

$u_k(i,j)$ je délka nejkratší cesty z i do j, která prochází pouze vrcholy $1, 2, \dots, k$.

1.4.25 Tvrzení. Platí

1. $\mathbf{U}_0$ je matice délek $\mathbf{A}$.
2. $\mathbf{U}_n$ je matice vzdáleností $\mathbf{U}$.
3. Matici $\mathbf{U}_{k+1}$ získáme z matice $\mathbf{U}_k$ takto:

$$u_{k+1}(i,j) = \min\{u_k(i,j), u_k(i,k+1) + u_k(k+1,j)\}.$$

Důkaz: První dvě vlastnosti jednoduše vyplývají z definice matic $\mathbf{U}_0$ a $\mathbf{U}_n$.

Třetí vlastnost dostaneme, když si uvědomíme, že nejkratší cesta z i do j, která vede pouze přes vrcholy $1, 2, \dots, k+1$ se buď vrcholu $k+1$ vyhne (a pak je délky $u_k(i,j)$), nebo vrcholem $k+1$ prochází a pak je délky $u_k(i,k+1) + u_k(k+1,j)$.

1.4.26 Floydův algoritmus.

Vstup: matice délek $\mathbf{A}$.

Výstup: matice vzdáleností $\mathbf{M} = \mathbf{U}$.

1. *[Inicializace]*
   $\mathbf{M} := \mathbf{A}$
2.
```
begin
    for k = 1, 2, ..., n do
        for i = 1, 2, ..., n do
            for j = 1, 2, ..., n do
            begin
                if M(i,j) > M(i,k) + M(k,j) then
                    M(i,j) = M(i,k) + M(k,j)
            end
end
```

1.4.27 Ukončení Floydova algoritmu je zaručeno tím, že vnější cyklus se provádí n-krát, tj. variant je k, které se roste od 1 do n.

Invariantem je 1.4.24 a vlastnost 3 z 1.4.25.

1.4.28 Huffmanův kód pro kompresi dat. Jsou dána data obsahující znaky z abecedy C a pro každý znak $c \in C$ je dána četnost $c.freq$ výskytu c v datech. Kódovat znaky můžeme buď slovy stejné délky; délka jednotlivého kódového slova je dána počtem znaků — je to nejmenší k takové, že $|C| \leq 2^k$. V takovém případě je délka komprimovaných dat rovna součinu počtu znaků a délky jednotlivého kódového slova.

Jinou možností je kódovat znaky slovy o nestejné délce. V případě kódových slov o nestejné délce je však třeba, aby žádné kódové slovo pro znak abecedy C nebylo prefixem jiného kódového slova. V tomto případě je délka dat po kompresi rovna

$$\sum_{c \in C} c.freq \cdot |w(c)|,$$

kde $w(c)$ je kódové slovo znaku c a $|w(c)|$ je jeho délka.

Každý kód si můžeme představit jako binární strom T, kde listy jsou ohodnoceny znaky abecedy C, hrany symbolem 0 nebo 1 a to tak, že ohodnocení cesty od kořene stromu k listu c je kódové slovo znaku c. Délka dat po kompresi je pak dána výrazem

$$B(T) = \sum_{c \in C} c.freq \cdot d_T(c),$$

kde $d_T(c)$ je hloubka listu c ve stromě T.

Huffmanův kód je binární kód nestejné délky jehož binární strom T má nejmenší možnou hodnotu $B(T)$.

1.4.29 Konstrukce Huffmanova kódu. Vstup: Máme dánu abecedu C, $n = |C|$, a četnosti $c.ferq$ jednotlivých znaků $c \in C$.
Výstup: Strom T optimálního binárního kódu.

1. Vytvoříme n jednoprvkových stromů T_c, každý kořen je označený $c; c.freq$; $Q = C$.

2. Dokud $|Q| \neq 1$, vybereme $x \in Q$ s nejmenší hodnotou $x.freq$ a $y \in Q$ s druhou nejmenší hodnotou $y.freq$. Do Q přidáme prvek z, položíme $z.freq := x.freq + y.freq$, a x, y odstraníme z Q. Vytvoříme strom T_z s kořenem z (označeným $z; z.freq$) takto: levý podstrom z je strom T_x, pravý podstrom je strom T_y (stromy T_x a T_y odstraníme).

3. Pro $Q = \{q\}$ je T_q binární strom, který určuje binární kód takto: každou hranu do levého následníka označ 0, do pravého následníka označ 1. Položime $T := T_q$.

1.4.30 Variant. Po každém průchodu bodem 2 má množina Q o jeden prvek méně. Tedy po $n - 1$ průchodech bodem 2 algoritmus skončí.

Obdobně bychom mohli říci, že při každém průchodu bodem 2 zmenšujeme počet stromů o jeden, a končíme v okamžiku, kdy máme pouze jeden strom.

1.4.31 Invariant. Nechť C je abeceda a $c.freq$, $c \in C$, jsou frekvence výskytů znaků v datech. Nechť x a y jsou dva znaky s nejmenšími frekvencemi. Vytvořme $C' = (C \setminus \{x, y\}) \cup \{z\}$, kde $z.freq = x.freq + y.freq$. Označme T' optimální strom (tj. strom s nejmenším $B(T')$) pro C'.

Pak strom T, který jsme dostali z T' nahrazením vrcholu z stromem s kořenem z, levým následníkem x a pravým následníkem y, je optimální strom pro C.

1.4.32 Myšlenka důkazu. Dá se dokázat, že kdykoli ze stromu T' pro abecedu C' vytvoříme strom T tak, že list z s $z.freq = x.freq + y.freq$ nahradíme výše popsaným stromem (kořen z, levý podstrom x, pravý podstrom y), tak

$$B(T) = B(T') + x.ferq + y.freq.$$

Nyní k dokončení důkazu potřebujeme vědět, že je vědy možné najít optimální kód pro abecedu C, takový, že v něm znaky x a y mají stejnou délku a liší se pouze v posledním bitu. A to říká následující lemma.

1.4.33 Lemma . Máme dánu abecedu C s frekvencemi $c.freq$. Nechť x a y jsou dva znaky s nejmenšími frekvencemi. Pak existuje optimální kód nestejné délky, kde kódová slova pro x a y mají stejnou délku a liší se pouze v posledním bitu.