Antud suunatud graaf $G = (V, E)$, iga serva $e \in E$ jaoks defineeritud tema *pikkus* $\ell(e)$.

Tee $v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} v_2 \xrightarrow{e_3} \cdots \xrightarrow{e_n} v_n$ *pikkus* on $\ell(e_1) + \ell(e_2) + \cdots + \ell(e_n)$.

Antud kaks tippu $u, v \in V$. Leida nendevaheline lühim tee.

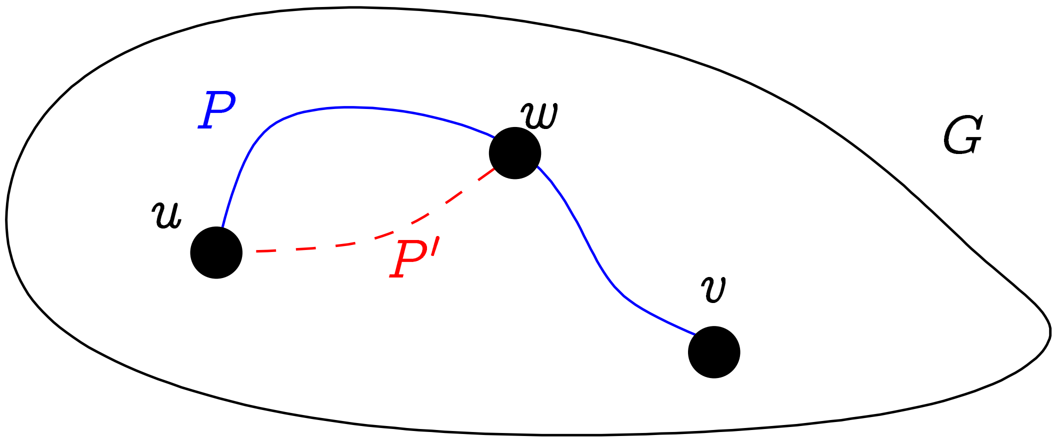
Tähistusi:

- Kui $v \in V$, siis Gv tähistagu kõigi selliste tippude w hulka, kuhu on tipust v serv ning $G^{-1}v$ kõigi selliste tippude w hulka, kust on tippu v serv.
- Kui $u, v \in V$, siis olgu $\ell(u, v)$ lühima u-st v-sse mineva serva pikkus. Kui u-st v-sse ei lähe ühtegi serva, siis olgu $\ell(u, v) = \infty$.
- Kui $u, v \in V$, siis olgu $d(u, v)$ lühima tee pikkus u-st v-sse. Kui u-st ei saa v-sse, siis $d(u, v) = \infty$.
- Kui $u, v \in V$ ja $i \in \mathbb{N}$, siis olgu $D_i(u, v)$ sellise lühima tee pikkus u-st v-sse, kus on ülimalt i serva. Kui selliseid teid ei ole, siis $D_i(u, v) = \infty$.

Ülesandel on optimaalne alamstruktuur:

Olgu P lühim tee tipust u tipuni v. Olgu w mingi tipp sellel teel. Siis tee P algusosa tipust u tipuni w on lühim tee u-st w-sse.

Tõepoolest, kui leiduks lühem tee P' u-st w-sse, siis võiks P algusosa asendada P'-ga ja saada P-st lühema tee u-st v-sse.



Järeldus: iga u, v, w jaoks $d(u,v) \leqslant d(u,w) + \ell(w,v)$.

Kehtib $D_0(u,v) \geqslant D_1(u,v) \geqslant D_2(u,v) \geqslant \cdots$. Peale selle, mingi i jaoks $D_i(u,v) = D_{i+1}(u,v) = \cdots = d(u,v)$.

Selline i leidub küll ainult siis, kui graafis pole *negatiivse pikkusega tsükleid*. Siis $i \leqslant |V| - 1$, sest lühim tee läbib iga tippu ülimalt ühel korral.

Kehtib (alati)

$$D_{i+1}(u,v) = \min\bigl(\{D_i(u,v)\} \cup \{\, D_i(u,w) + \ell(w,v) \,:\, w \in G^{-1}v\}\bigr),$$

seega saab negatiivse pikkusega tsüklite mitteesinemisel leida tippude kaugusi tipust u järgmisel viisil:

Olgu D ja DD massiivid, mis on indekseeritud graafi $G = (V, E)$ tippudega.

$D[u] := 0$
for all $v \in V \setminus \{u\}$ **do** $D[v] := \infty$
for all $v \in Gu$ **do** $D[v] := \ell(u, v)$
repeat
    $DD := D$
    for all $v \in V$ **do**
        $d := DD[v]$
        for all $w \in G^{-1}v$ **do**
            $d := \min(d, DD[w] + \ell(w, v))$
        $D[v] := d$
until $D = DD$
return D

Invariant: peale i-ndat iteratsiooni (reas 11) on $D[v] = D_i(u, v)$.

Ridades 7–9 toimuvat miinimumi leidmist võib ka otse väljal $D[v]$ teha, s.t. abimuutujat d kasutamata. Siis omistamist $D[v] := DD[v]$ pole vaja teha, sest need on niikuinii võrdsed (5. rea tõttu).

```
D[u] := 0
for all v ∈ V\{u} do D[v] := ∞
for all v ∈ Gu do D[v] := ℓ(u, v)
repeat
    DD := D
    for all v ∈ V do
        for all w ∈ G^{-1}v do
            D[v] := min(D[v], DD[w] + ℓ(w, v))
until D = DD
return D
```

Siin 6. ja 7. rida korraldavad lihtsalt tsükli, mis graafi kõik servad korra läbi käib.

Ta on korraldatud nii, et käime kõigepealt läbi servade võimalikud lõpptipud v ja seejärel kõik servad, mis selles tipus lõppevad.

Tsüklit üle servade võib ka teisiti korraldada, näiteks käia kõigepealt läbi kõik võimalikud algtipud ja seejärel kõik servad, mis sellest tipust algavad.

Levinud graafiesitusviiside korral (tippude lihtahel ja iga tipu juures temast algavate servade lihtahel) võib selline korraldus efektiivsem olla.

$D[u] := 0$

for all $v \in V \setminus \{u\}$ **do** $D[v] := \infty$

for all $v \in Gu$ **do** $D[v] := \ell(u, v)$

repeat

    $DD := D$

    for all $w \in V$ **do**

        for all $v \in Gw$ **do**

            $D[v] := \min(D[v], DD[w] + \ell(w, v))$

until $D = DD$

return D

Kui meil on negatiivse pikkusega tsükleid, siis sellist i-d ei leidu, et $D_i(u,v) = D_{i+1}(u,v) = D_{i+2}(u,v) = \cdots$.

Siis pole isegi $d(u,v)$ defineeritud.

Kuidas negatiivse pikkusega tsüklite olemasolu avastada?

Kui ei ole negatiivse pikkusega tsükleid, siis on iteratsioone, kus midagi muutub, ülimalt $|V| - 1$.

Teemegi ülimalt $|V|$ iteratsiooni ja kui viimasel iteratsioonil midagi muutus, siis anname veateate.

$D[u] := 0$
for all $v \in V \setminus \{u\}$ **do** $D[v] := \infty$
for all $v \in Gu$ **do** $D[v] := \ell(u, v)$
$i := 0$
repeat
    $DD := D$
    for all $w \in V$ **do**
        for all $v \in Gw$ **do**
            $D[v] := \min(D[v], DD[w] + \ell(w, v))$
    $i := i + 1$
until $D = DD$ **or** $i = |V|$
if $D = DD$ **then return** D
error „negatiivse pikkusega tsükkel“

Hakkama saab ka ilma massiivita DD:

$D[u] := 0$
for all $v \in V \setminus \{u\}$ **do** $D[v] := \infty$
for all $v \in Gu$ **do** $D[v] := \ell(u, v)$
$i := 0$
repeat
  $muutus := \mathsf{false}$
  for all $w \in V$ **do**
    for all $v \in Gw$ **do**
      if $D[v] > D[w] + \ell(w, v)$ **then**
        $D[v] := D[w] + \ell(w, v)$
        $muutus := \mathsf{true}$
  $i := i + 1$
until $\neg muutus$ **or** $i = |V|$
if $\neg muutus$ **then return** D
error „negatiivse pikkusega tsükkel“

Tsükliinvariant on antud juhul (tsükli lõpus, real 13):

Peale i-ndat iteratsiooni on $D[v]$ minimaalse pikkusega tee mingite u-st v-sse viivate teede hulgas. Seejuures see hulk sisaldab kõiki teid u-st v-sse, milles on ülimalt i serva.

Peale eelviimast iteratsiooni sisaldab see hulk seega kõiki teid u-st v-sse, mis läbivad ülimalt $(|V|-1)$-t serva. Muuhulgas siis ka lühimat teed u-st v-sse.

Esitatud algoritm on *Bellman-Ford*i algoritm.

Keerukus:

Välimist tsüklit (read 5–13) itereeritakse kuni $|V|$ korda.

Selle sees on sisemine tsükkel (read 7–11) üle kõigi servade.
Selle tsükli keha (read 9–11) võtab konstantse aja.

Kokku seega $\Theta(|V| \cdot |E|)$.

Toodud algoritm leiab lühimate teede pikkused tipust u teistesse tippudesse.

Kuidas leida/salvestada teid ise?

Iga tipu v jaoks salvestame talle eelneva tipu (senileitud) lühimal teel u-st v-sse. Kasutame selleks massiivi π (indekseeritud graafi tippudega).

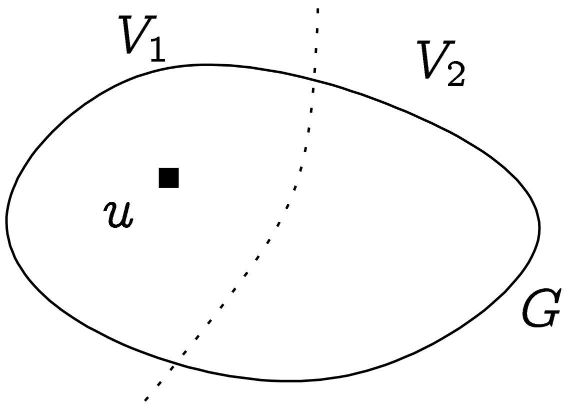
Lühim tee u-st v-sse on siis (ümberpööratult) $v, \pi[v], \pi[\pi[v]], \pi[\pi[\pi[v]]], \ldots, u$.

Massiivid D ja π rahuldavad iga tipu v jaoks, kus $\pi[v]$ on defineeritud: $D[v] = D[\pi[v]] + \ell(\pi[v], v)$.

$D[u] := 0$
for all $v \in V \setminus \{u\}$ **do** $D[v] := \infty$
for all $v \in Gu$ **do** $D[v] := \ell(u, v)$; $\pi[v] := u$
$i := 0$
repeat
    $muutus := \mathsf{false}$
    for all $w \in V$ **do**
        for all $v \in Gw$ **do**
            if $D[v] > D[w] + \ell(w, v)$ **then**
                $D[v] := D[w] + \ell(w, v)$
                $\pi[v] := w$
                $muutus := \mathsf{true}$
    $i := i + 1$
until $\neg muutus$ **or** $i = |V|$
if $\neg muutus$ **then return** (D, π)
error „negatiivse pikkusega tsükkel“

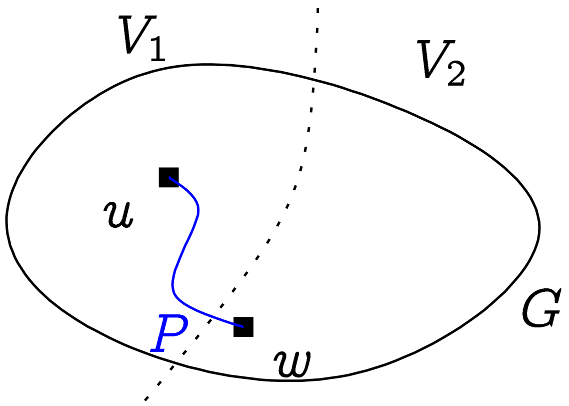
Eeldame nüüd, et negatiivse pikkusega servi ei leidu. Siis saab lühimaid teid u-st teistesse tippudesse kiiremini leida.

Oletame, et me oleme juba leidnud V_1-e kuuluvate tippude jaoks lühimate teede pikkused u-st nendesse tippudesse.



Samuti oletame, et ükski V_1-e kuuluv tipp pole u-st kaugemal kui ükski V_2-e kuuluv tipp.

Olgu w u-le lähim tipp osas V_2. Olgu P lühim tee u-st w-sse.



Siis P kõik tipud peale w asuvad osas V_1. Kui veel mõni P tippudest asuks osas V_2, siis oleks see tipp u-le lähemal kui w.

Tahame seda w-d leida. Me eemaldaks ta V_2-st ja lisaks V_1-e. Eelmise slaidi all toodud invariant jääks siis kehtima.

Leidmaks seda tippu w:

Me peame vaatama ainult selliseid ahelaid u-st V_2-e tippudesse, mille kõik tipud, peale viimase, asuvad V_1-s.

Olgu antud V_1 ja V_2, samuti olgu iga $v \in V_1$ jaoks antud $D[v]$ — lühima tee pikkus u-st v-sse.

Vaatame kõiki servi (v, w), kus $v \in V_1$ ja $w \in V_2$. Leiame $D[v] + \ell(v, w)$. Leiame v ja w, mille korral see summa minimaalne on.

Lisame w V_1-e ning võtame $D[w] := D[v] + \ell(v, w)$ ja $\pi[w] := v$.

Selle w kiiresti leidmiseks:

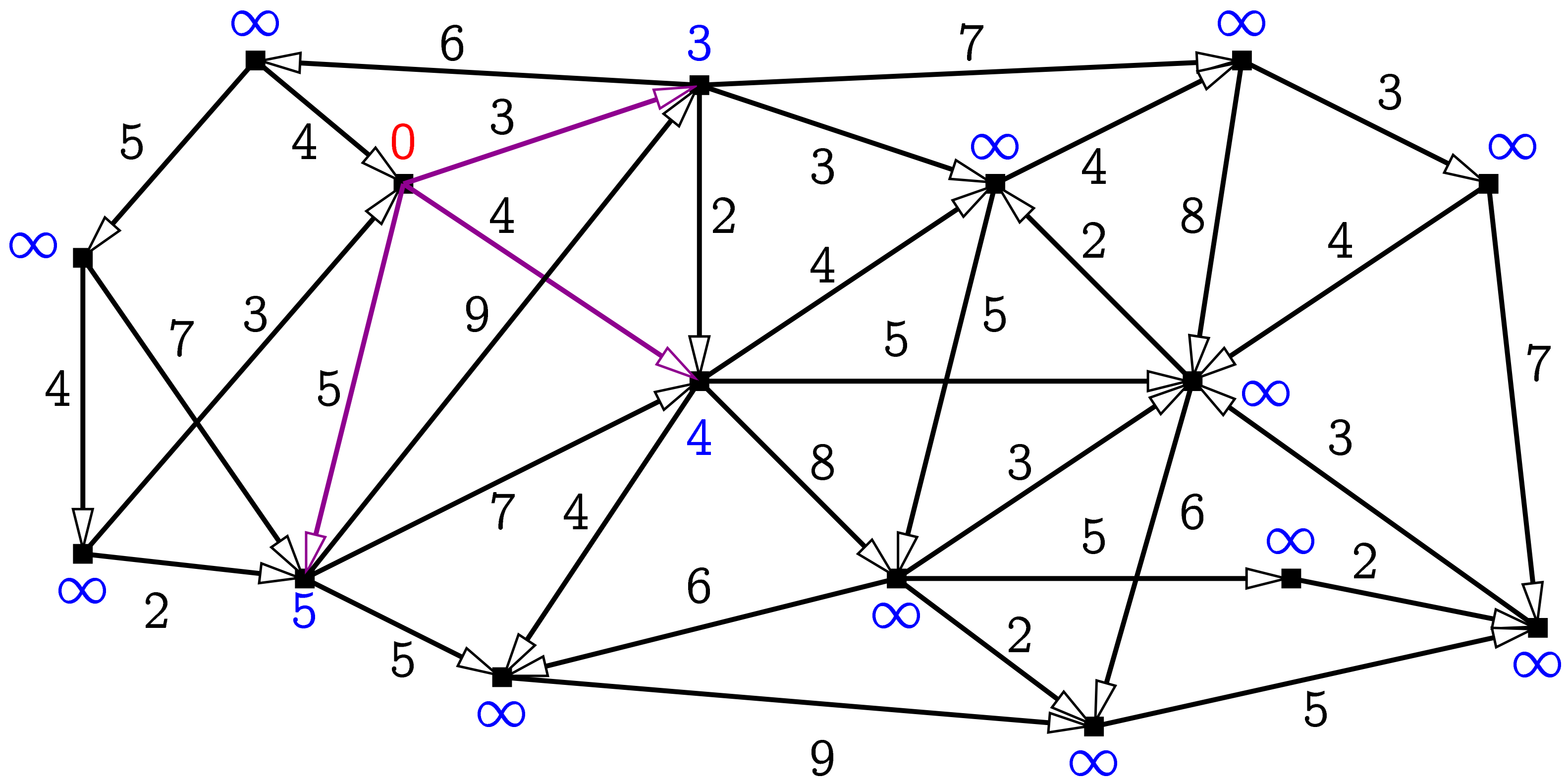
Iga $w \in V_2$ jaoks olgu tema kaal $\delta(w)$ defineeritud kui

$$\delta(w) := \min_{v \in V_1} D[v] + \ell(v, w) \quad .$$

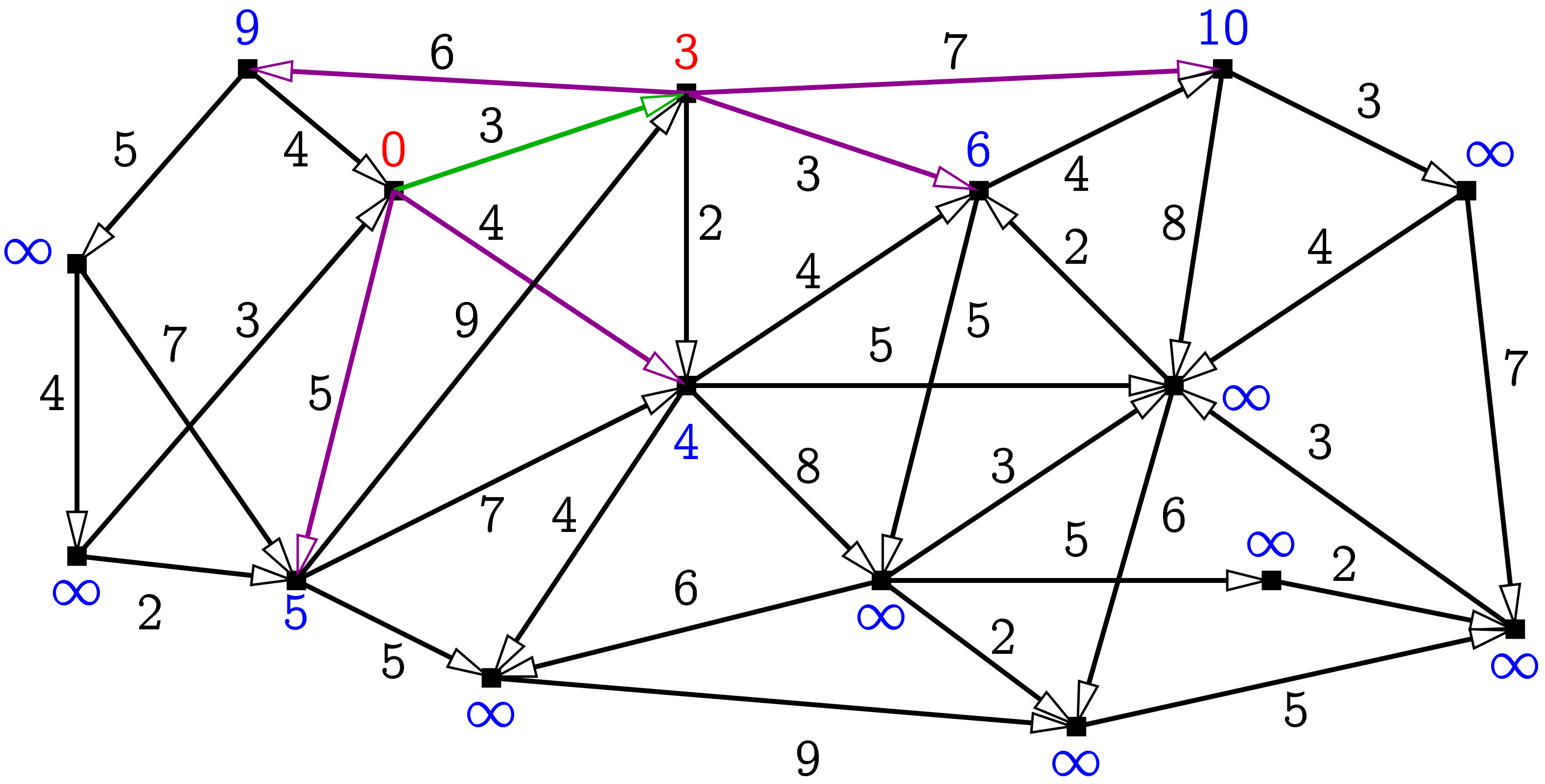
Hoiame V_2 elemente kuhjas (võtmeks on kaal $\delta(\cdot)$). Siis saab minimaalse $\delta(\cdot)$-ga tippu lihtsalt leida.

Kui me tipu w viime V_2-st V_1-e, siis võivad w naabertippude kaalud väheneda.

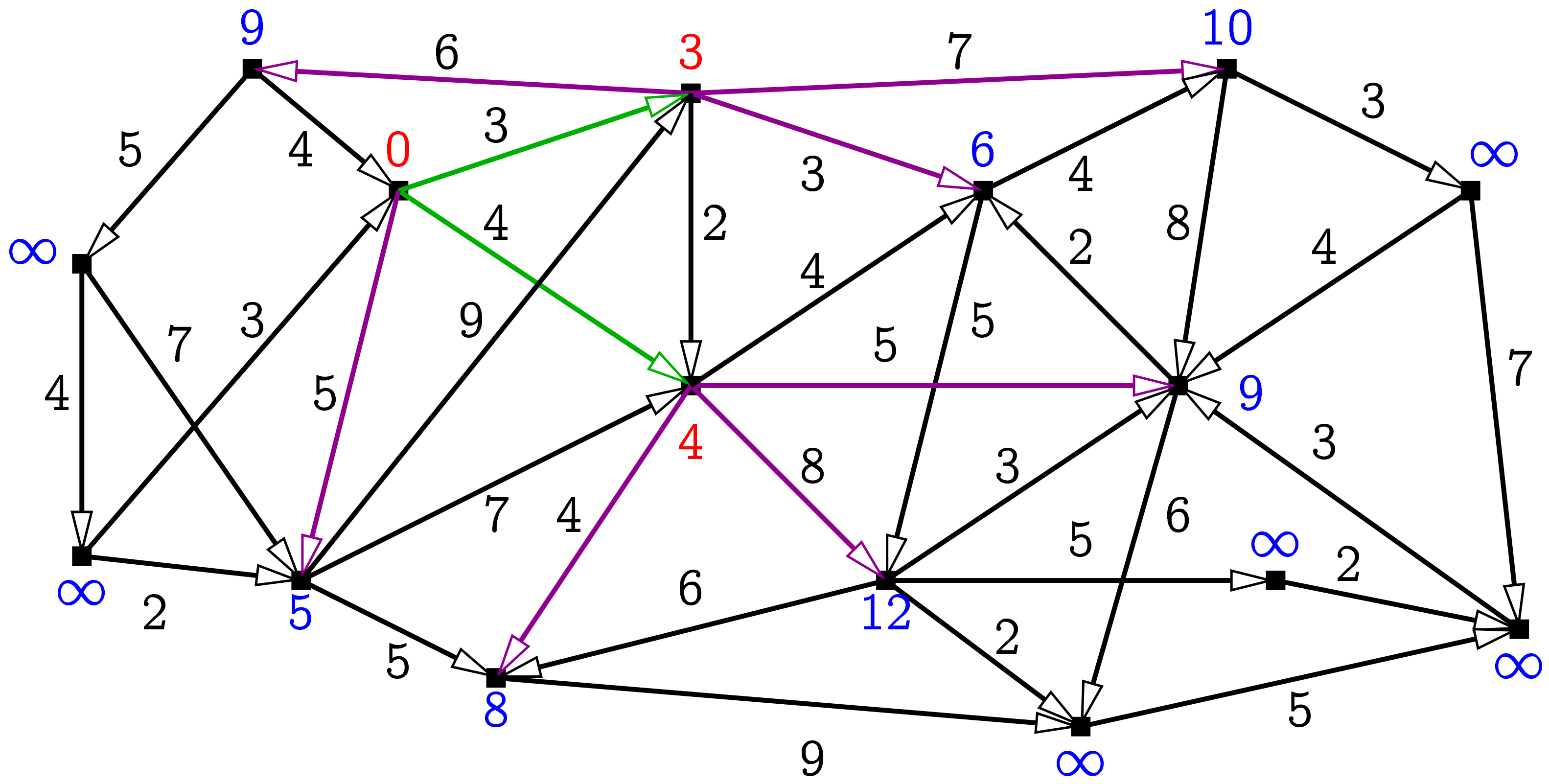
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



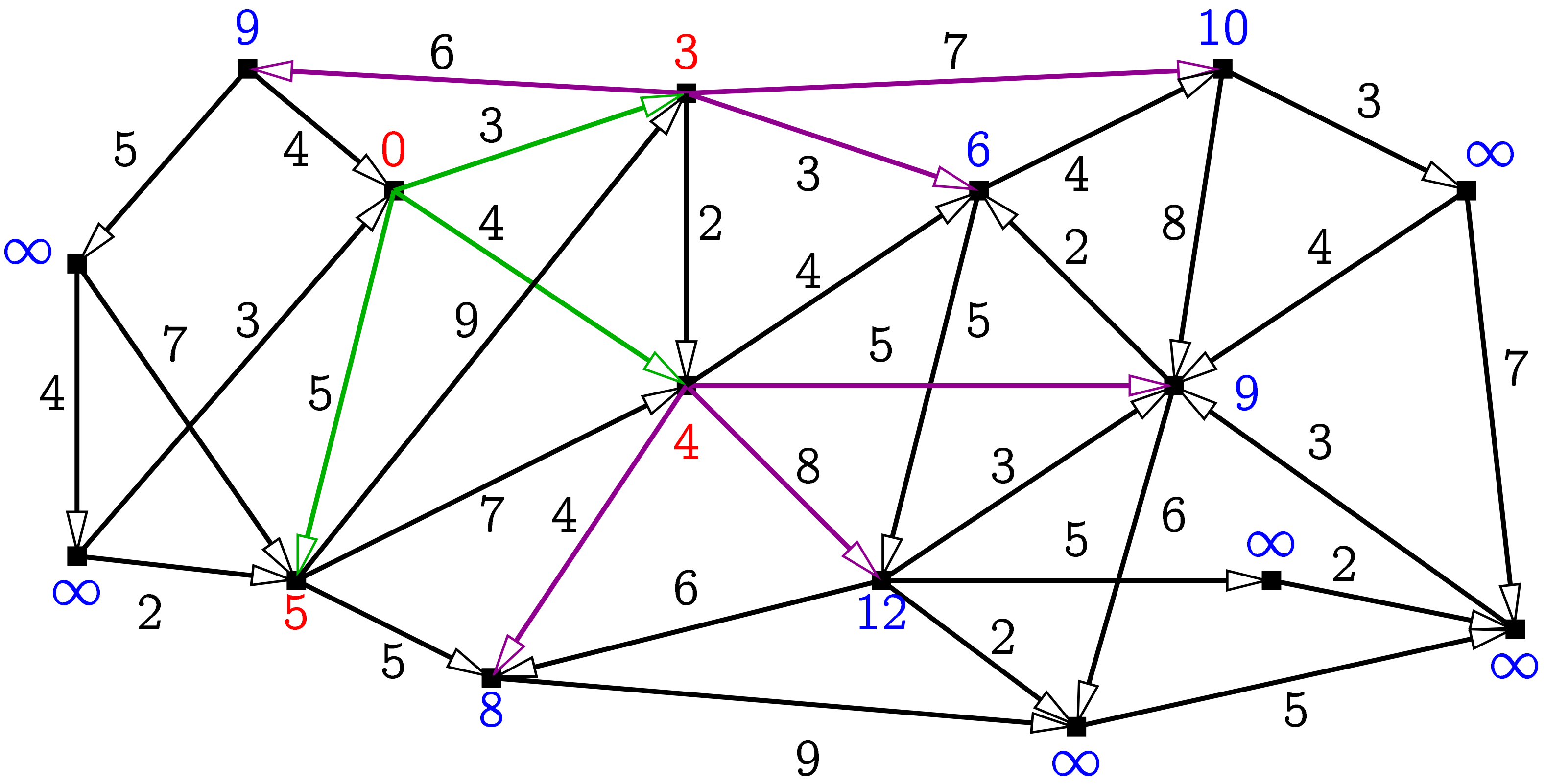
Näide: (V_1-e kuuluva tipu kaugus u-st,V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib,π-ga määratud servad)



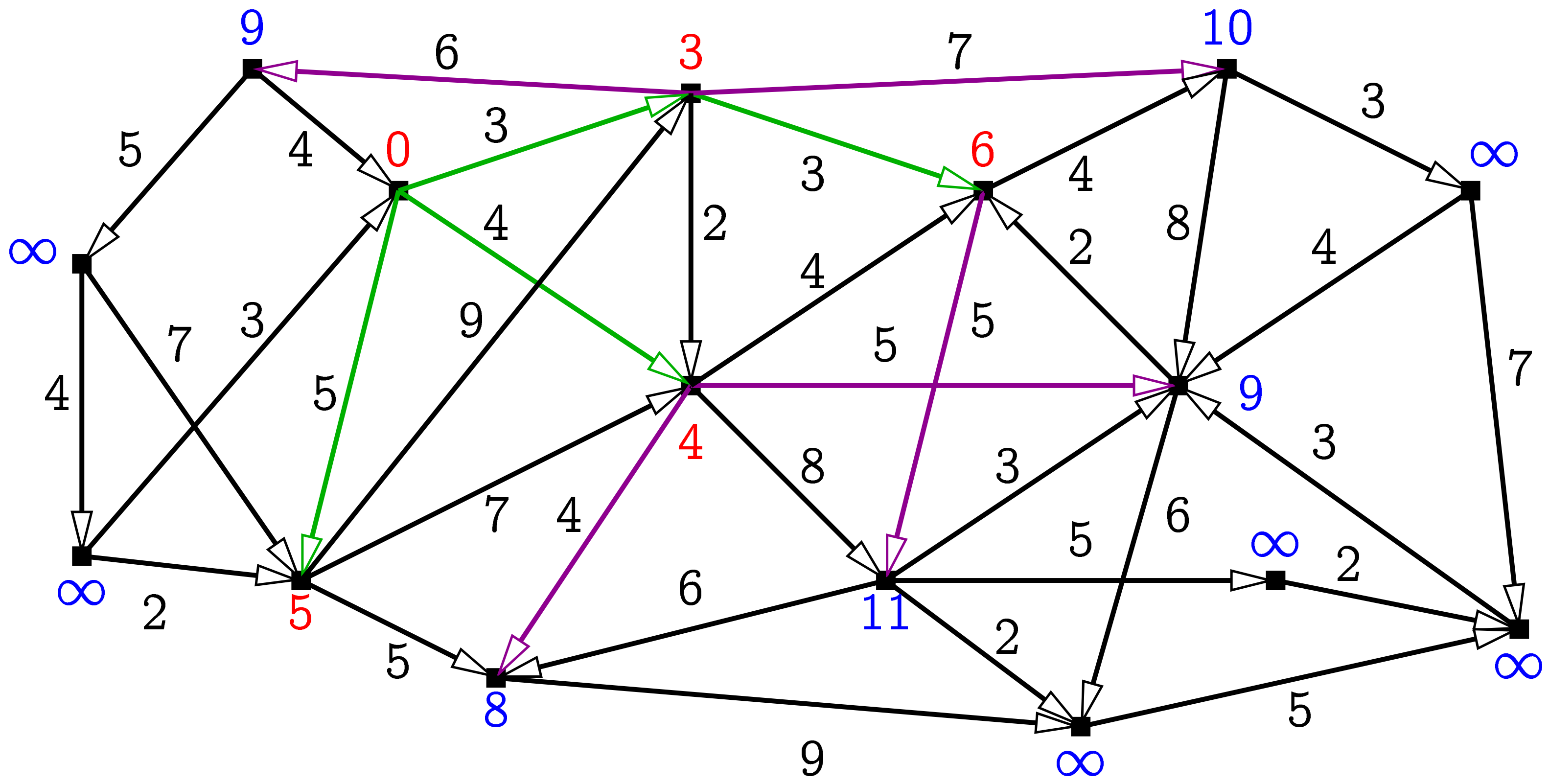
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



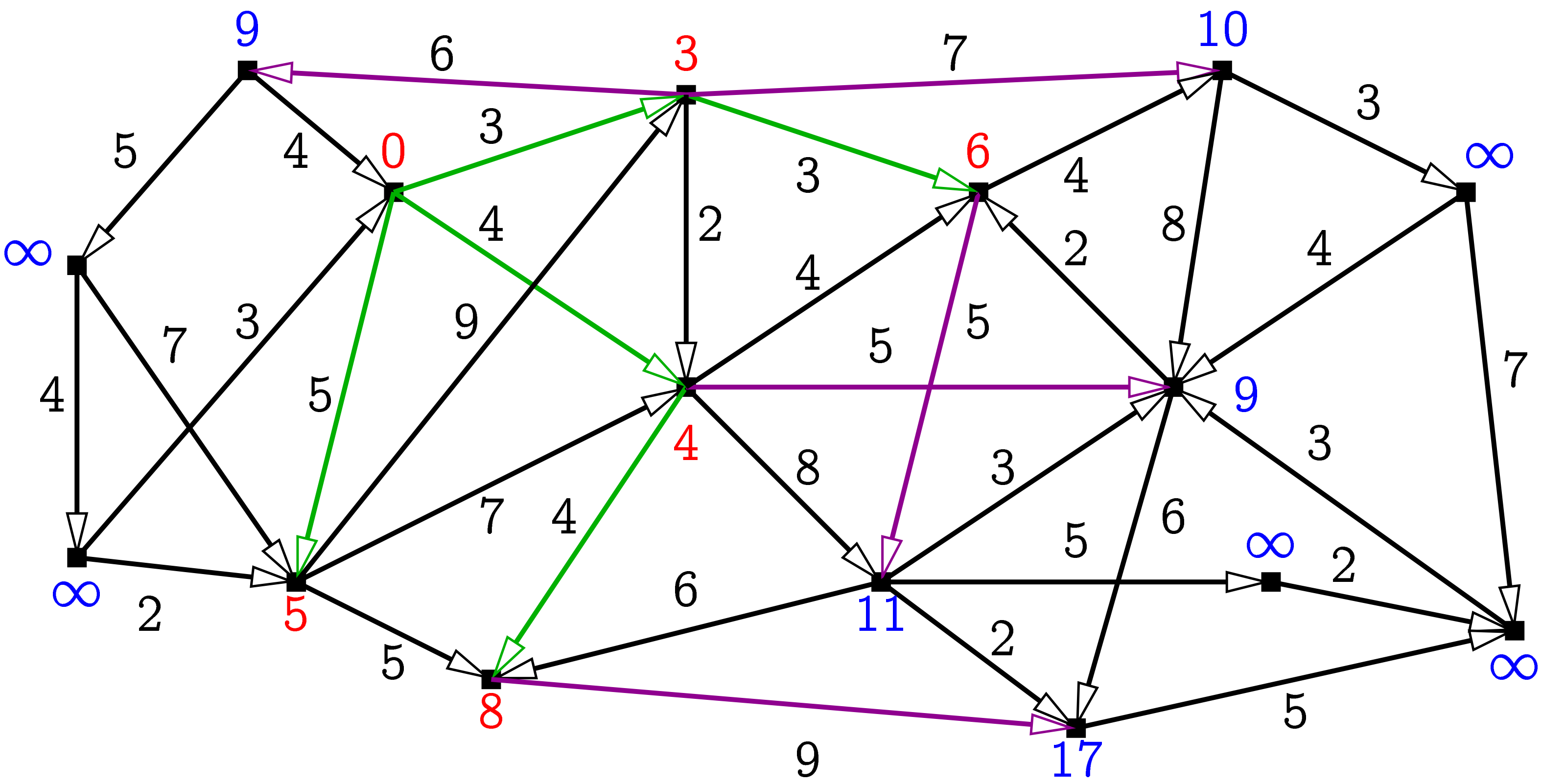
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



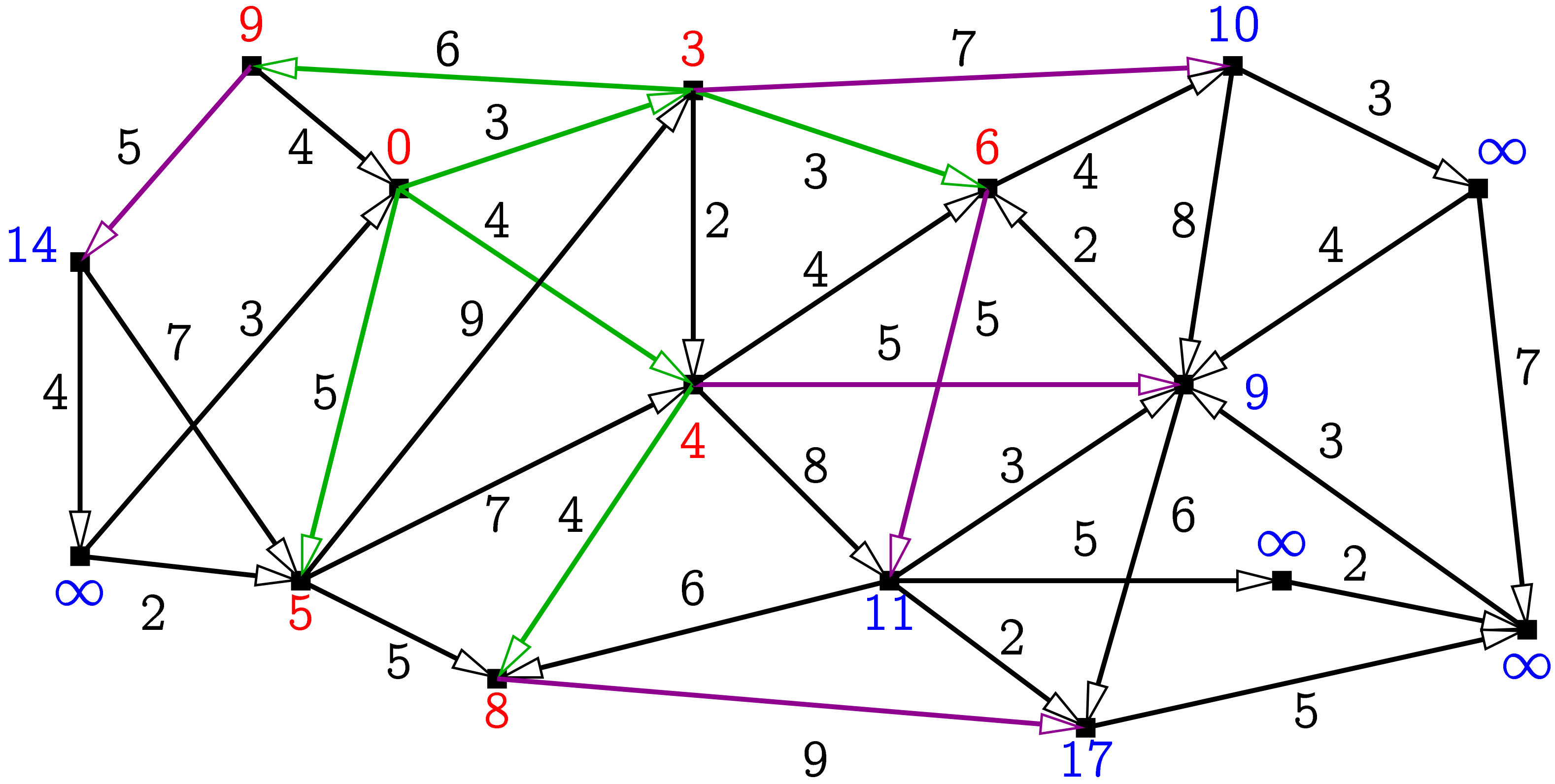
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



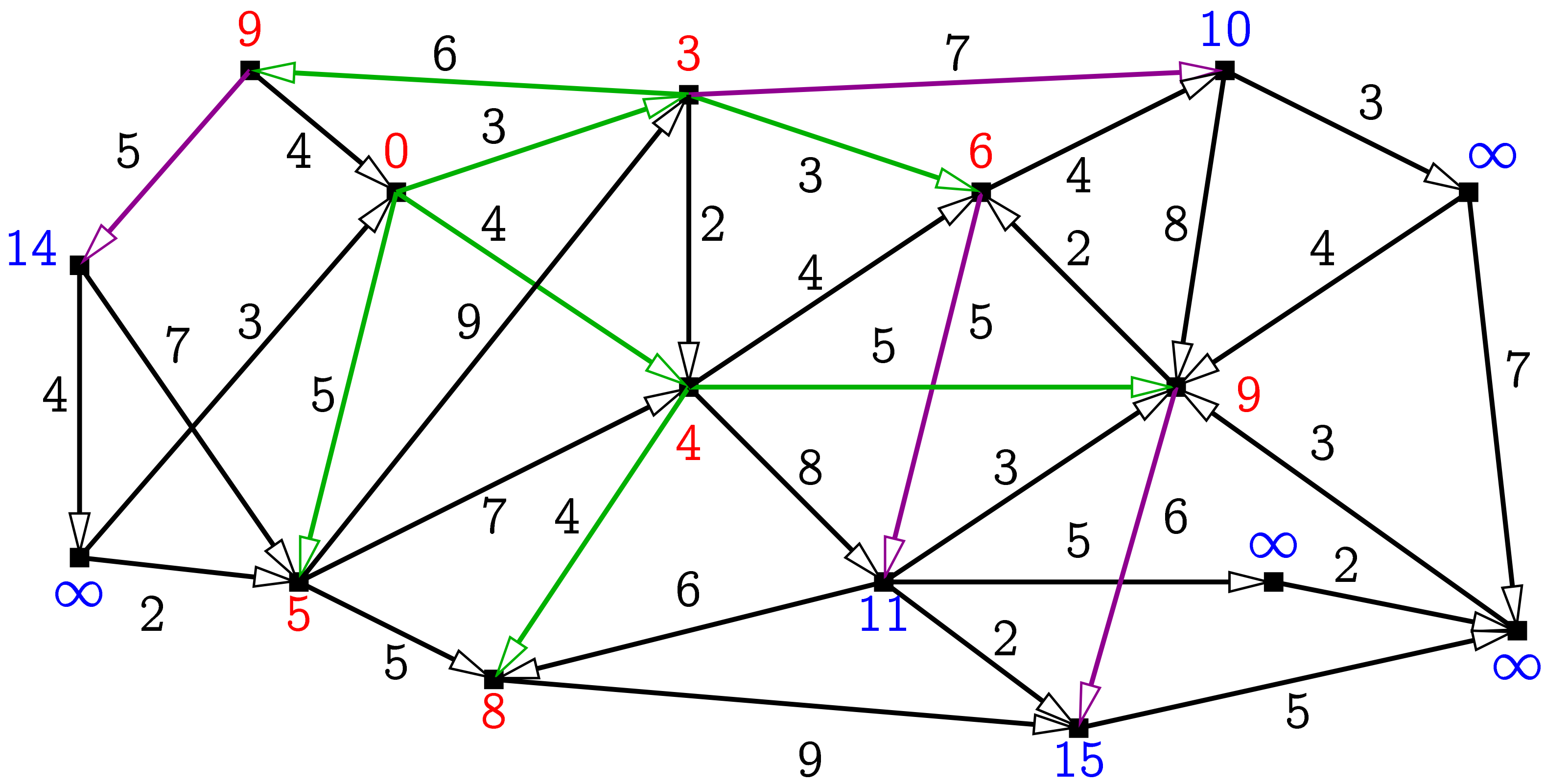
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



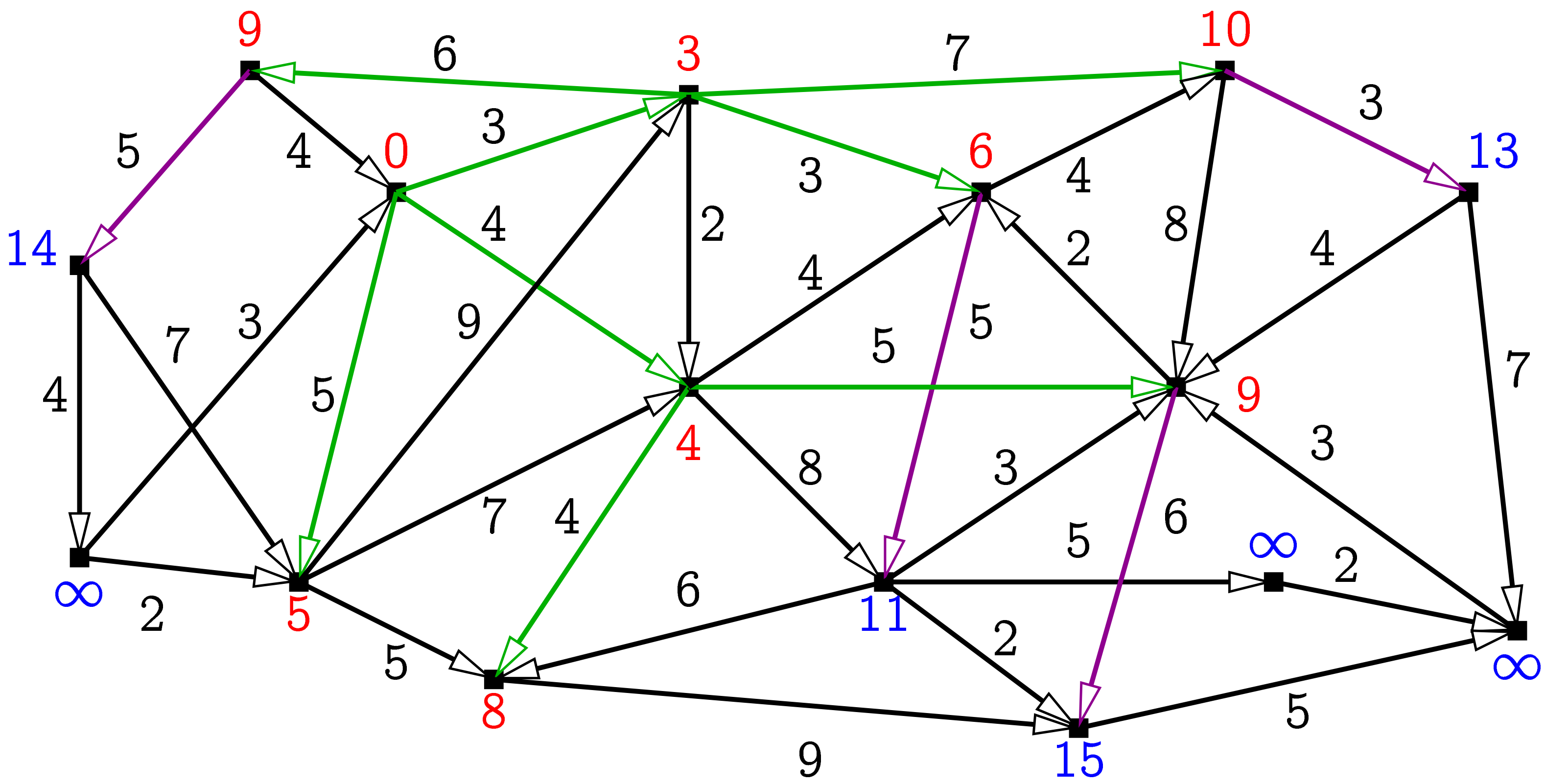
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



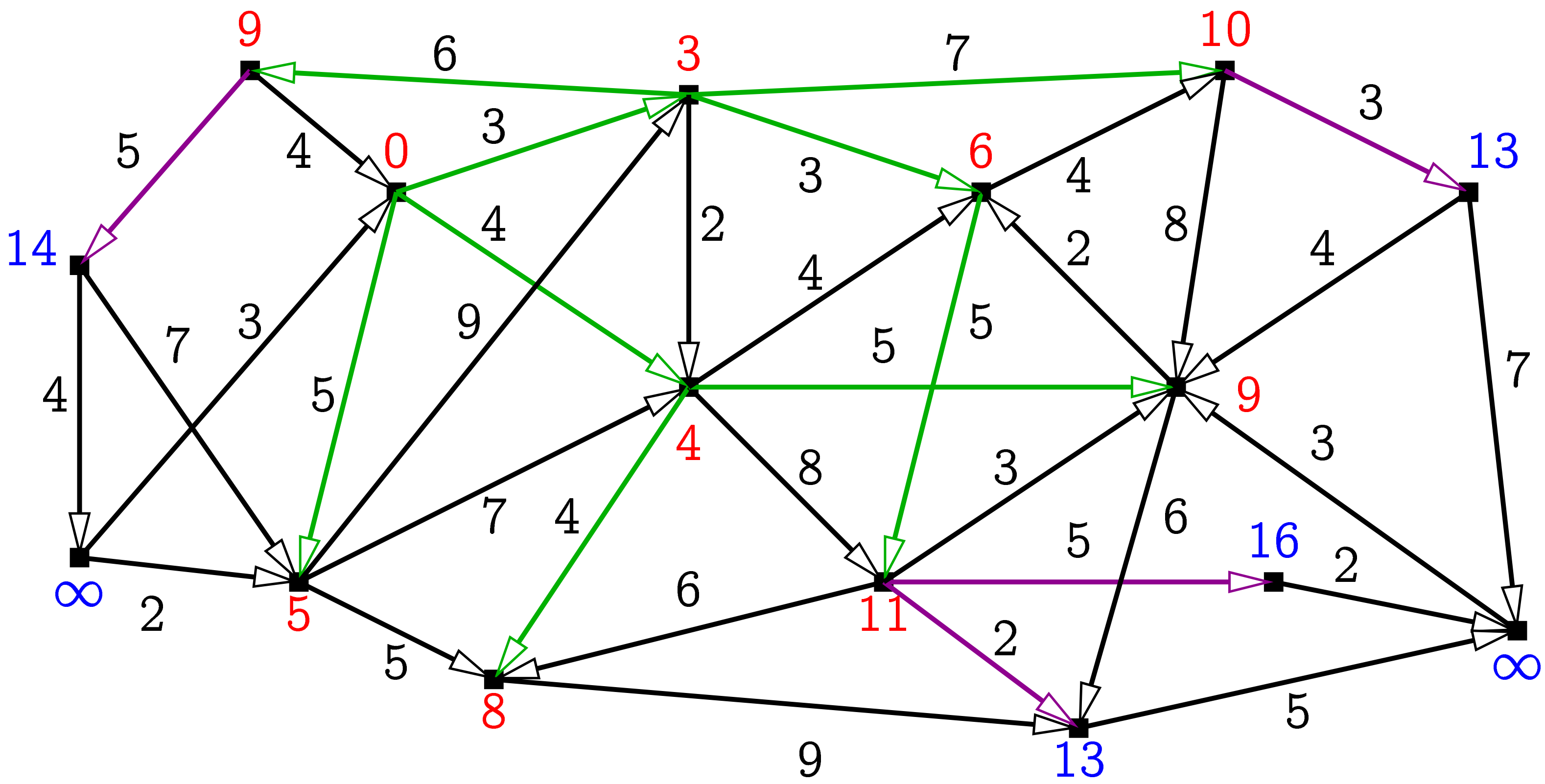
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



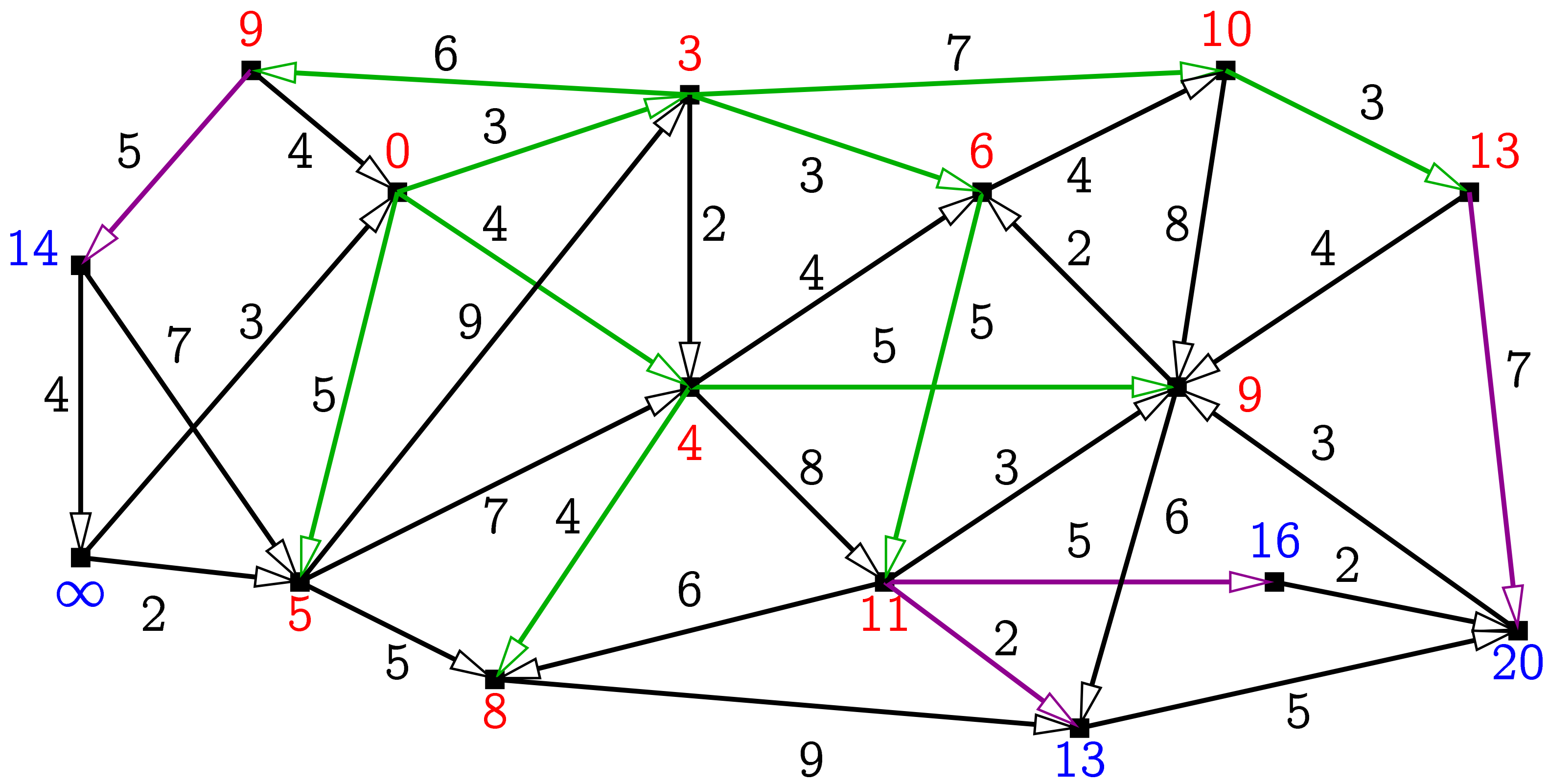
Näide: (V_1-e kuuluva tipu kaugus u-st,V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib,π-ga määratud servad)



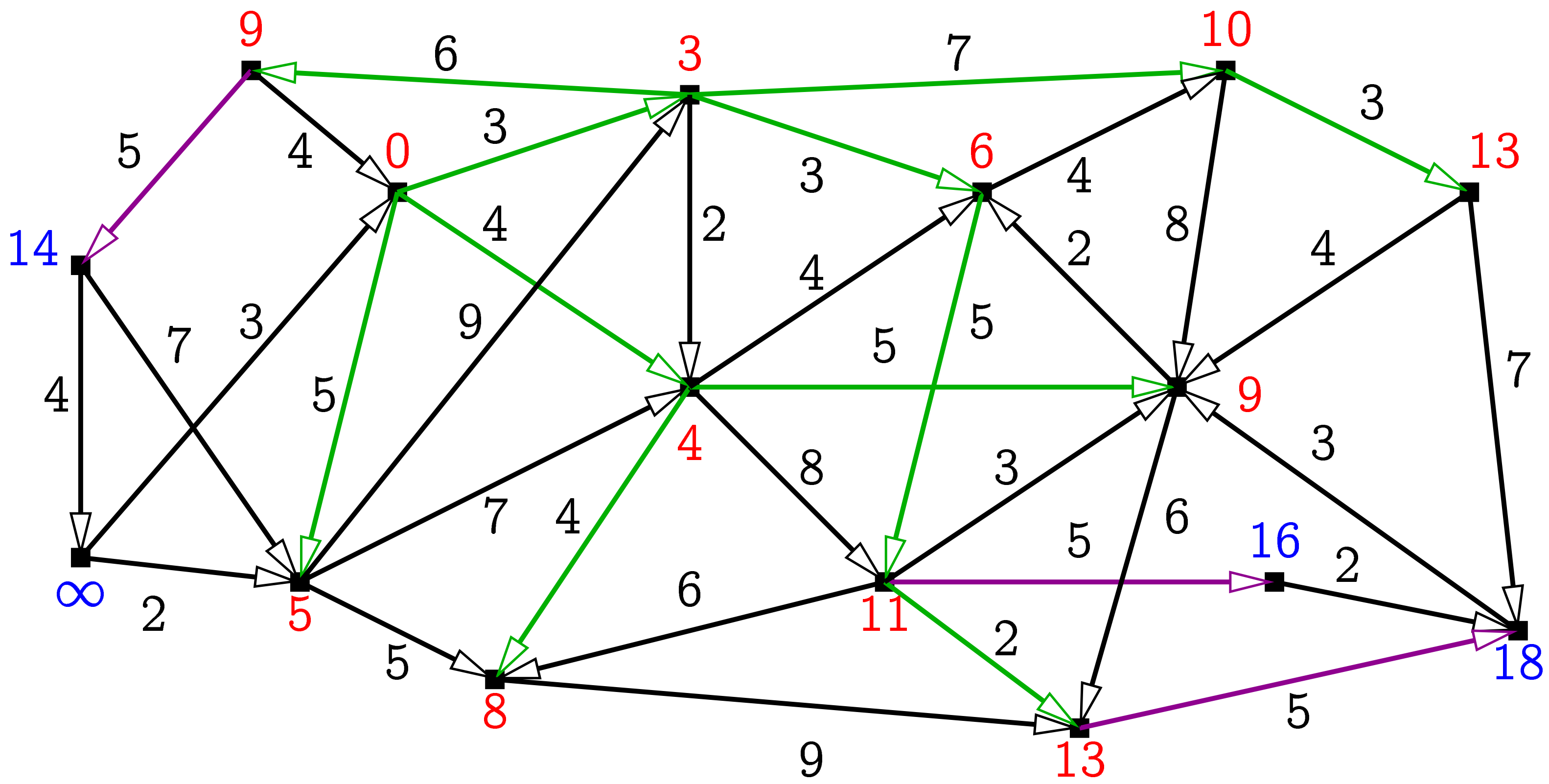
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



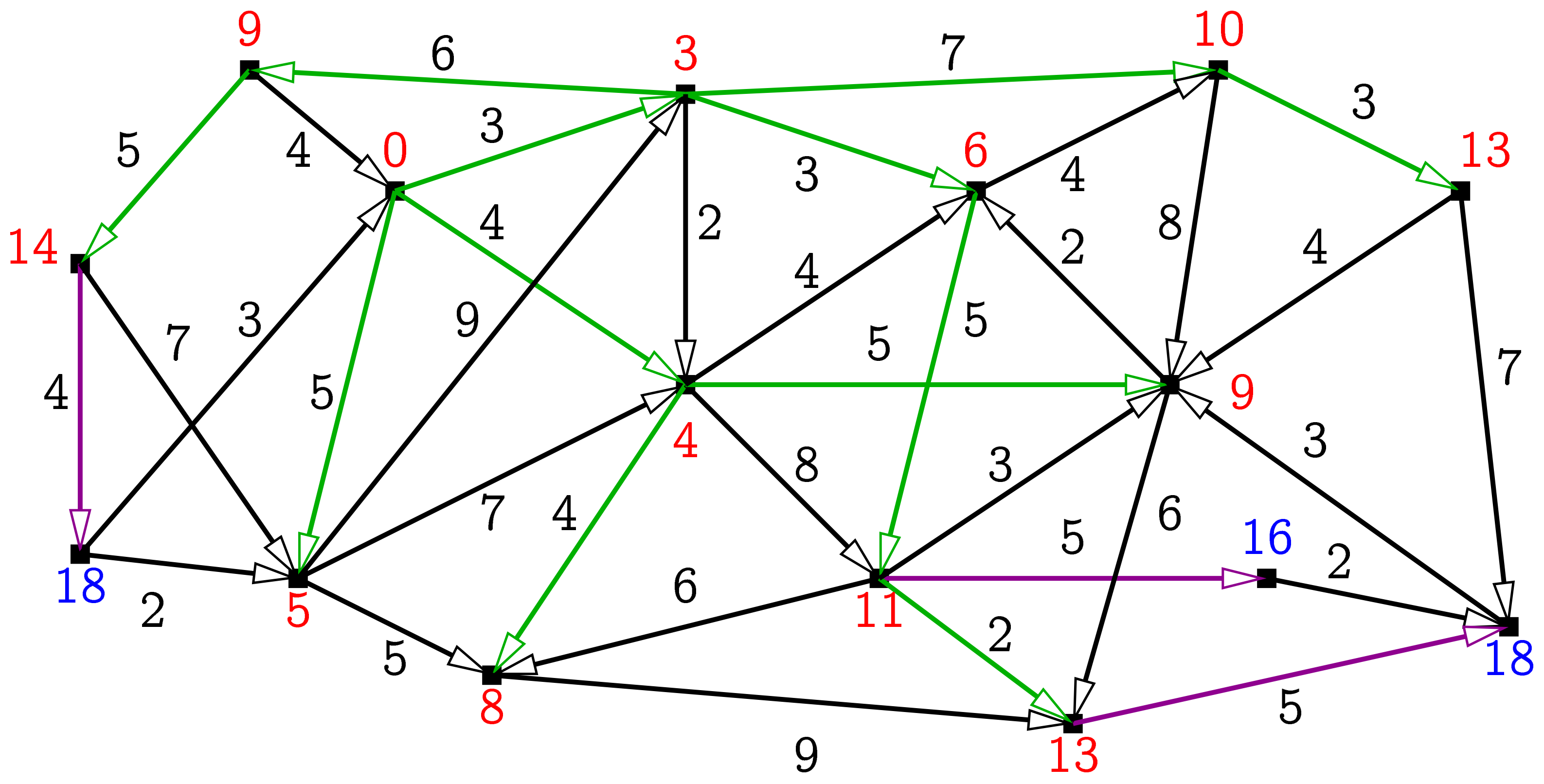
Näide: (V_1-e kuuluva tipu kaugus u-st,V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib,π-ga määratud servad)



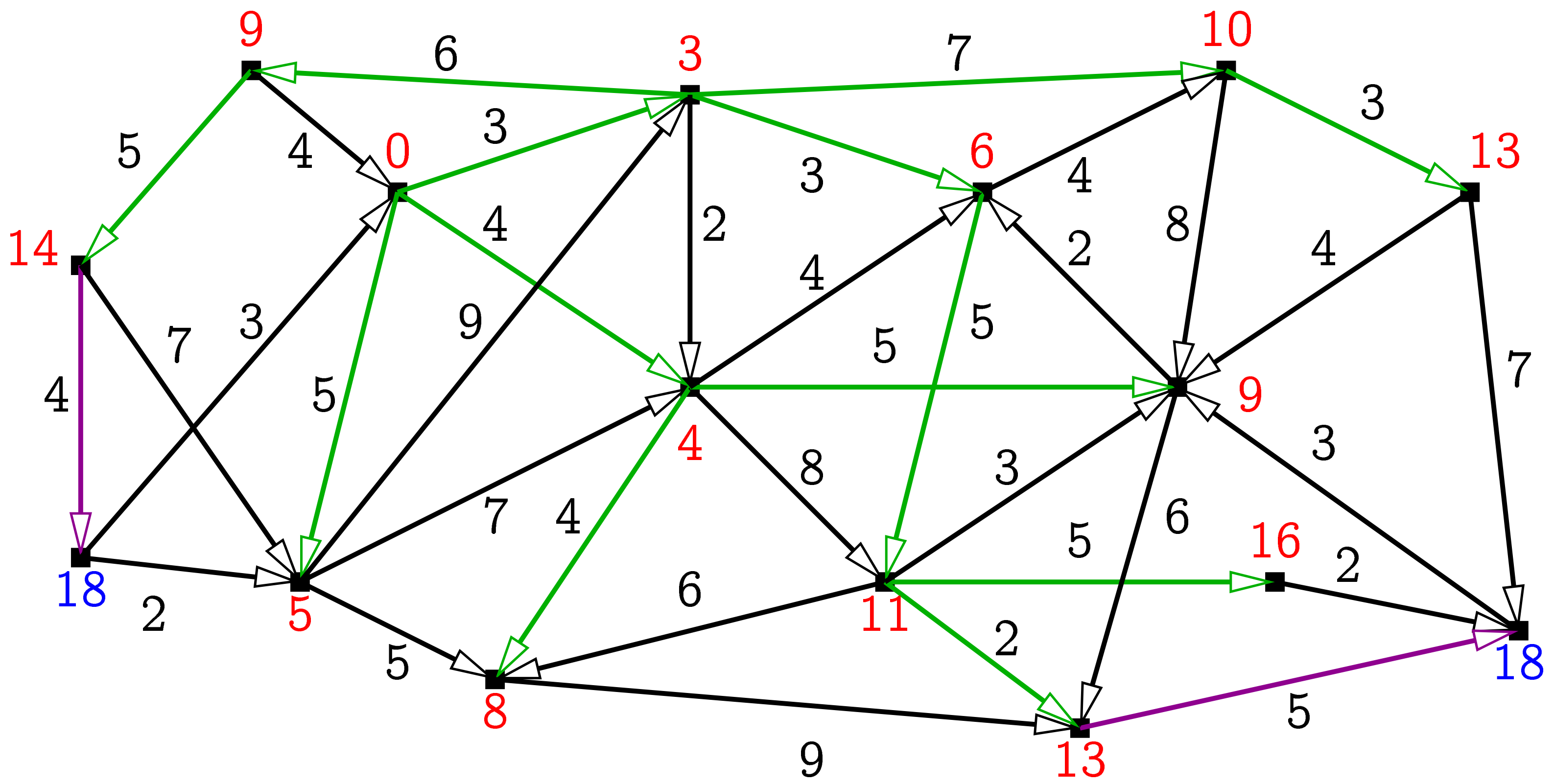
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



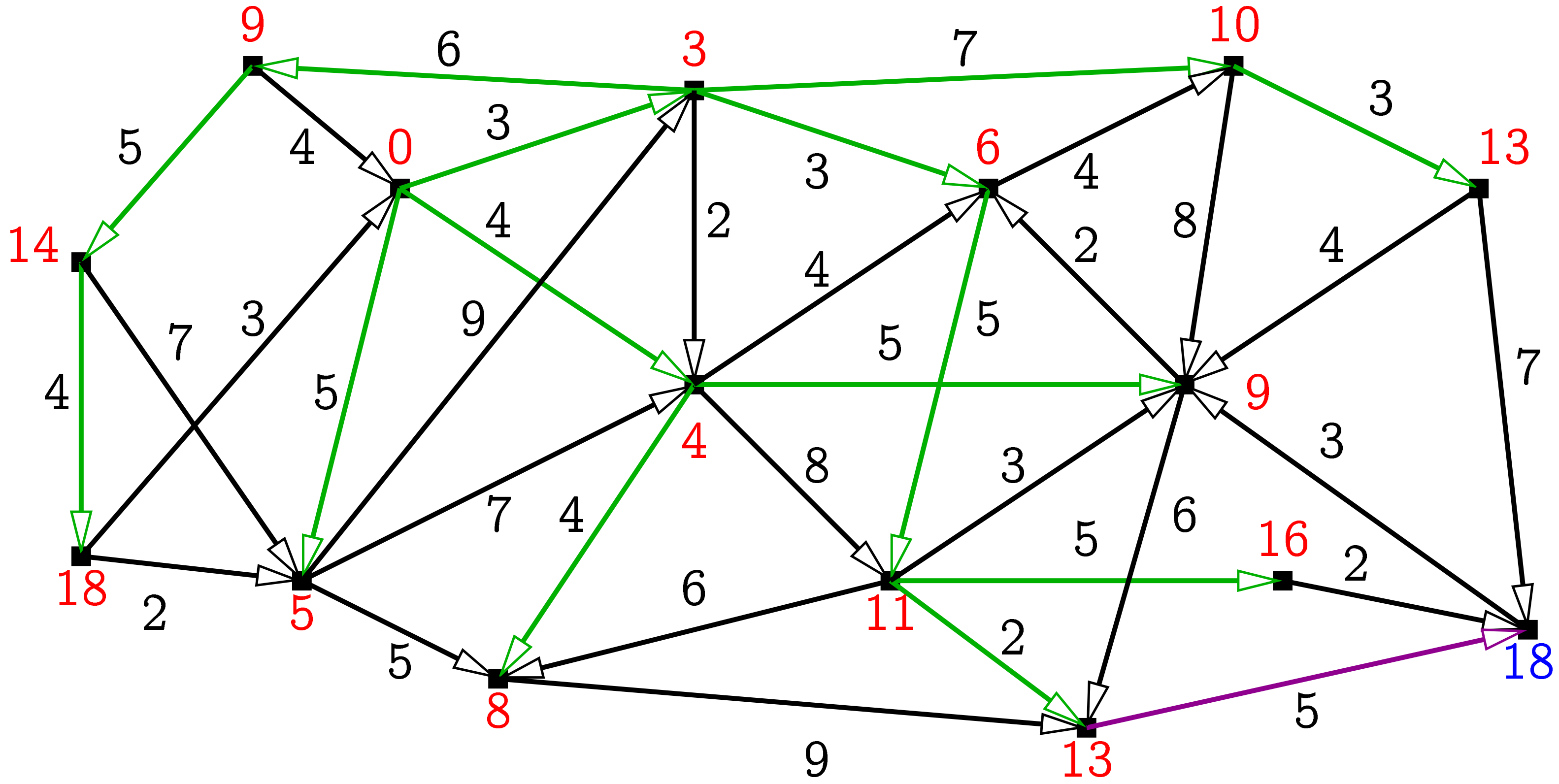
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



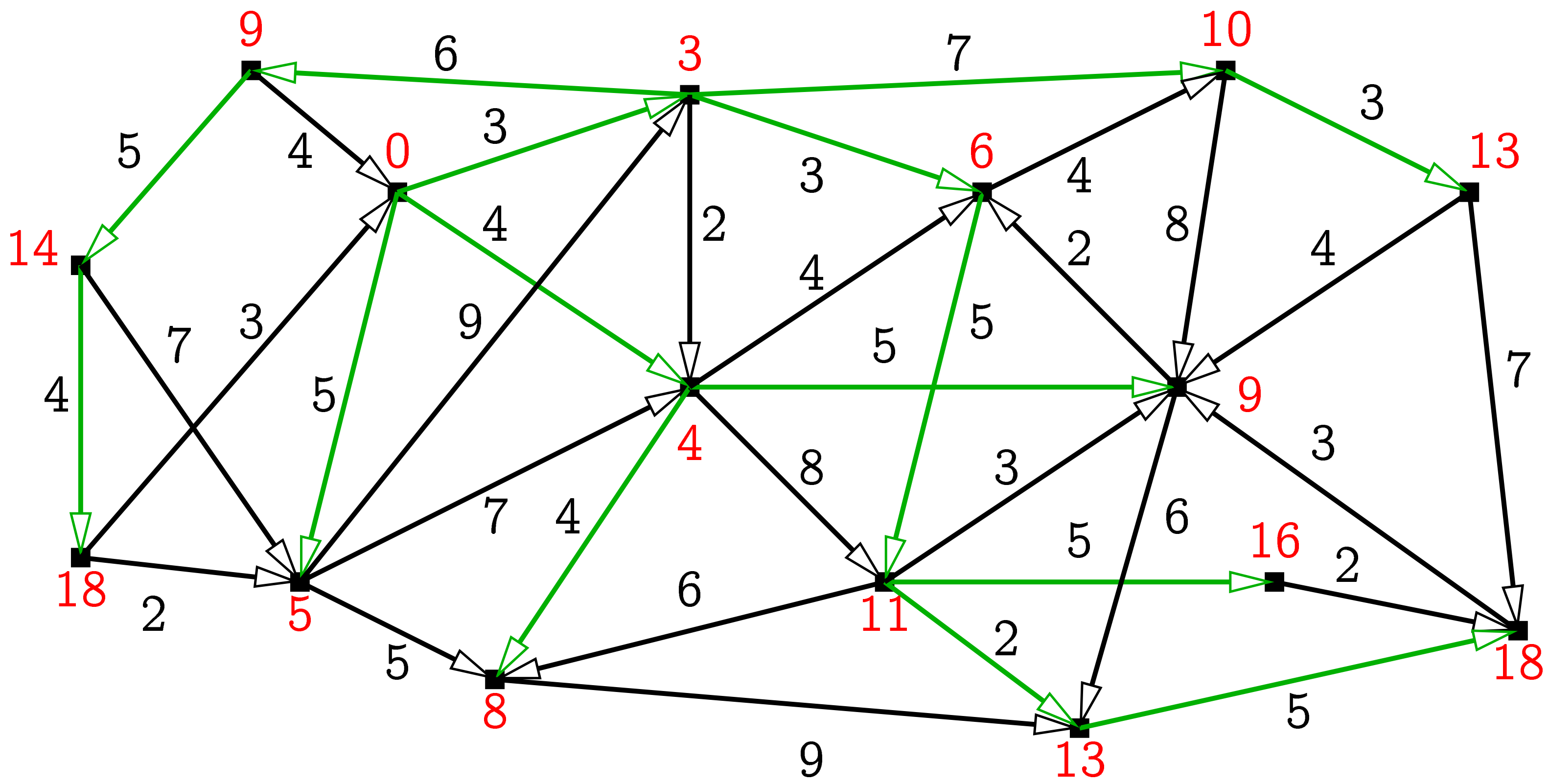
Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



Näide: (V_1-e kuuluva tipu kaugus u-st, V_2-e kuuluva tipu kaal ja serv, mis selle kaalu realiseerib, π-ga määratud servad)



Algoritmi realiseerides võib tipu kauguse u-st (V_1-e kuuluvate tippude jaoks) ja tipu kaalu (V_2-e kuuluvate tippude jaoks) kasutada sama massiivi D.

Kui me tõstame tipu w hulgast V_2 hulka V_1 ja seejuures tema naabertippude kaalusid vähendame, siis ei ole meil tarvis eksplitsiitselt kontrollida, kas naabertipp kuulub V_2-e. Kui me saame naabertipu kaalu vähendada, siis oli tema kaal suurem kui w kaal ja seega kuulus ta V_2-e.

Olgu Q mingi kuhi (kahend- või Fibonacci). Lühimate teede leidmise algoritm on siis (nimetatakse *Dijkstra* algoritmiks):

```
D[u] := 0
for all v ∈ V\{u} do D[v] := ∞
for all v ∈ Gu do D[v] := ℓ(u, v); π[v] := u
Q := kuhjasta(V\{u})        --- D[·] järgi
while Q ei ole tühi do
    w ⇐ Q
    if D[w] = ∞ then break
    for all v ∈ Gw do
        if D[v] > D[w] + ℓ(w, v) then
            D[v] := D[w] + ℓ(w, v)
            π[v] := w
            vii_üles(Q, v)
return (D, π)
```

Keerukus:

Ridu 5–7 täidetakse $O(|V|)$ korda. Reas 6 on min. kaaluga elemendi võtmine kuhjast, keerukus $O(\log |V|)$, kõigi iteratsioonide peale kokku seega $O(|V| \log |V|)$.

Ridu 8–12 täidetakse $O(|E|)$ korda. Reas 12 on kuhjaparandusoperatsioon, keerukus $O(\log |V|)$ või $O(1)$, sõltuvalt kuhjast. Kõigi iteratsioonide peale kokku seega $O(|E| \log |V|)$ või $O(|E|)$.

Dijkstra algoritmi kogukeerukus on siis

- $O((|V| + |E|) \log |V|)$, kui kasutame kahendkuhja;
- $O(|V| \log |V| + |E|)$, kui kasutame Fibonacci kuhja.

Ülesanne: antud graaf $G = (V, E)$ koos servade pikkustega, nii et negatiivse pikkusega tsükleid ei ole. Leia kõikvõimalike $u, v \in V$ jaoks $d(u, v)$.

Kui negatiivse pikkusega servi ei ole, siis saame kasutada iga tipu jaoks Dijkstra algoritmi. Keerukus $O(|V|^2 \log |V| + |V||E|)$.

Kui on ka negatiivse pikkusega servi, siis Bellman-Fordi algoritmi kasutamine iga tipu jaoks oleks keerukusega $O(|V|^2|E|)$.

Järgnevas vaatame (üldiselt) efektiivsemaid algoritme kui Bellman-Ford iga tipu jaoks.

Tähistusi:

- $\mathrm{Mat}_{X\times Y}(K)$ — kõigi maatriksite hulk, mille read on indekseeritud hulga X elementidega, veerud hulga Y elementidega ja maatriksi elemendid on hulgast K.
- $\mathbb{R}_\infty$ — hulk $\mathbb{R}\cup\{\infty\}$.
- Kui $L \in \mathrm{Mat}_{X\times Y}(K)$, $i \in X$ ja $j \in Y$, siis $[L]_{ij}$ tähistagu maatriksi L elementi reas i ja veerus j.

Vaatame maatriksit $L \in \mathrm{Mat}_{V\times V}(\mathbb{R}_\infty)$, kus

$$[L]_{uv} = \begin{cases} 0, & \text{kui } u = v \\ \ell(u,v), & \text{kui } u \neq v \end{cases}.$$

Siis $[L]_{uv}$ näitab lühima sellise tee pikkust tipust u tippu v, kus on ülimalt üks serv.

Olgu $W_i \in \mathrm{Mat}_{V\times V}(\mathbb{R}_\infty)$ selline, et $[W_i]_{uv} = D_i(u,v)$. Siis $W_1 = L$ ning W_0 on selline, kus peadiagonaalil on 0-d ja mujal ∞-d.

Neg. pikkusega tsüklite puudumise tõttu $W_{|V|-1} = W_{|V|} = W_{|V|+1} = \cdots$. Meil tuleb see maatriks leida.

Vastavalt varemnäidatud seosele D_{i+1} ja D_i vahel:

$$[W_{i+1}]_{uv} = \min_{w \in V}([W_i]_{uw} + [L]_{wv}) \quad .$$

kombineeri_maatriksid(A, B) on

```
for all u ∈ V do
    for all v ∈ V do
        C[u, v] := ∞
        for all w ∈ V do
            C[u, v] := min(C[u, v], A[u, w] + B[w, v])
return C
```

Siis $W_{i+1} =$ kombineeri_maatriksid(W_i, L).

kombineeri_maatriksid keerukus on $O(|V|^3)$.

initsialiseeri_L on

for all $u \in V$ **do**
  for all $v \in V$ **do**
    $L[u,v] := \ell(u,v)$
  $L[u,u] := 0$
return L

Siis tippude kauguste maatriksi võib leida järgmiselt:

$L :=$ initsialiseeri_L; $W_1 := L$
for $i := 2$ **to** $|V|-1$ **do**
  $W_i :=$ kombineeri_maatriksid(W_{i-1}, L)
return $W_{|V|-1}$

Keerukus $O(|V|^4)$, pole parem kui Bellman-Ford igale tipule.

Algebraline struktuur $(R, \overline{+}, \bar{\cdot}, \overline{0}, \overline{1})$ on *poolring*, kui

- $(R, \overline{+}, \overline{0})$ on kommutatiivne monoid;
- $(R, \bar{\cdot}, \overline{1})$ on monoid;
- $\overline{0}$ on korrutamise nullelement, s.t. $\overline{0} \bar{\cdot} r = r \bar{\cdot} \overline{0} = \overline{0}$ iga $r \in R$ jaoks;
- $\overline{+}$ distributeerub $\bar{\cdot}$ suhtes mõlemalt poolt. S.t. iga $a, b, c \in R$ jaoks

$$a \bar{\cdot} (b \overline{+} c) = a \bar{\cdot} b \overline{+} a \bar{\cdot} c$$

$$(a \overline{+} b) \bar{\cdot} c = a \bar{\cdot} c \overline{+} b \bar{\cdot} c \quad .$$

Lause. Olgu R mingi poolring. Siis $\mathrm{Mat}_{V\times V}(R)$-s on maatriksite korrutamine assotsiatiivne.

Lause. $(\mathbb{R}_\infty, \min, +, \infty, 0)$ on poolring.

Selles poolringis $W_{i+1} = W_i \cdot L$. Seega $W_i = L^i$.

kombineeri_maatriksid kujutas endast just üle selle poolringi defineeritud maatriksite korrutamist.

Meie ülesandeks on leida L-i küllalt suur aste (vähemalt $|V|-1$). Assotsiatiivsuse tõttu võib seda leida L-i korduvalt ruutu tõstes.

$L :=$ initsialiseeri_L; $W := L$
for $i := 1$ **to** $\lceil \log |V| \rceil$ **do**
    $W :=$ kombineeri_maatriksid(W, W)
return W

Keerukus: $O(|V|^3 \log |V|)$, mis on parem kui Bellman-Ford iga tipu jaoks, kui $|E| = \Omega(|V| \log |V|)$.

Oleme leidnud lühimate teede pikkused tippude vahel. Aga kuidas leida tegelikke teid?

Iga kahe tipu $u, v \in V$ jaoks oleks tarvis leida tipp w — eelviimane tipp lühimas tees u-st v-sse. S.t. otsime maatriksit $\Pi \in \mathrm{Mat}_{V \times V}(V)$.

Olgu tipp u fikseeritud. Olgu $D[v]$ lühima tee pikkus u-st v-sse. Näitame, kuidas leida massiiv π, nii et $\pi[v]$ oleks eelviimane tipp lühimas tees u-st v-sse.

Maatriksi Π leidmiseks kutsume siis seda protseduuri välja iga $u \in V$ jaoks.

Moodustame graafi G_u järgmiselt:

- G_u tippudeks on G tipud;
- G_u-s on serv w-st v-sse parajasti siis, kui G-s on serv w-st v-sse ja kui $D[v] - D[w] = \ell(w, v)$.

Graafis G_u on iga tee u-st v-sse mingi graafi G lühim tee u-st v-sse.

Graafi G_u laiuti / sügavuti / mingil muul viisil läbides leiame teed u-st teistesse tippudesse.

Olgu J mingi dünaamiline järjend. Järgmine protseduur leiab π.

for all $v \in V$ **do** $v.\textit{läbitud} := \mathsf{false}$

$J := \emptyset$; $J \Leftarrow u$; $u.\textit{läbitud} := \mathsf{true}$

while J ei ole tühi **do**

  $w \Leftarrow J$

  for all $v \in G_u w$ **do**

    if $\neg v.\textit{läbitud}$ **then**

      $v.\textit{läbitud} := \mathsf{true}$

      $\pi[v] := w$

      $J \Leftarrow v$

Keerukus:

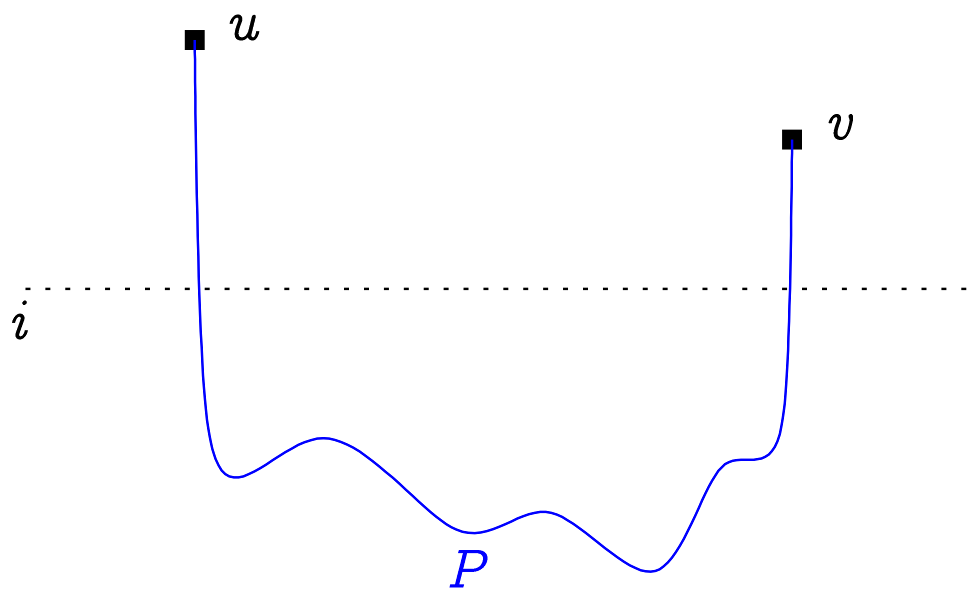
- G_u moodustamiseks: $O(|V| + |E|)$.
- π leidmiseks $O(|V|+|E|)$, kui dünaamilise järjendi operatsioonid on konstantse keerukusega.

Kokku seega $O(|V| + |E|)$. Kuna $|V|$ on $O(|E|)$, siis on π leidmise keerukus ka $O(|E|)$.

Kogu Π leidmine vajab seega aega $O(|V||E|)$.

Loeme nüüd, et graafi tippudeks on naturaalarvud $1, 2, \ldots, n$.

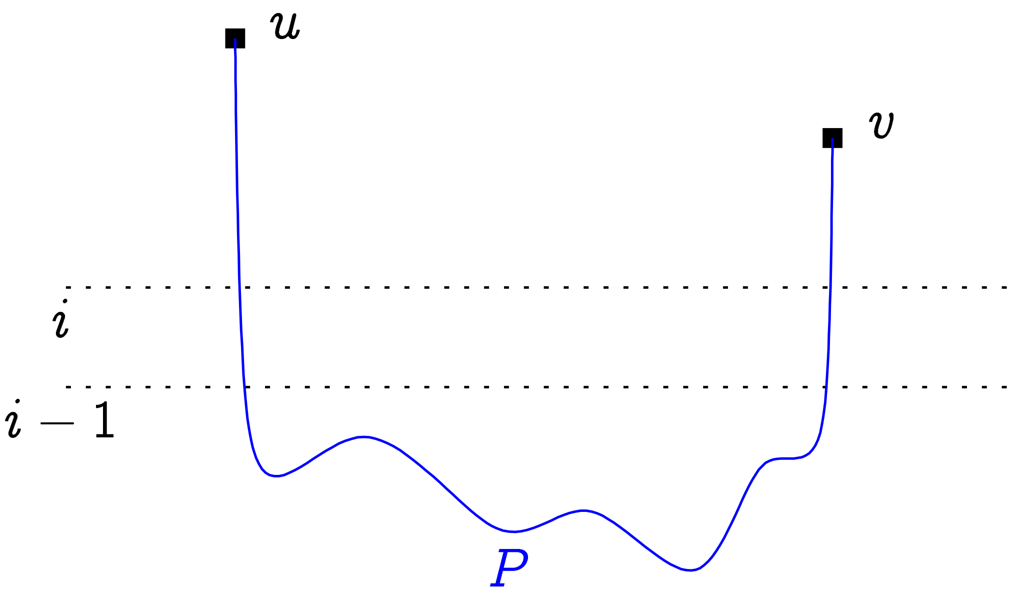
Iga $u, v, i \in V$ jaoks olgu $d^i(u, v)$ lühima sellise tee pikkus u-st v-sse, kus kõik tipud (peale u ja v) kuuluvad hulka $\{1, \ldots, i\}$. Siis $d(u, v) = d^n(u, v)$.



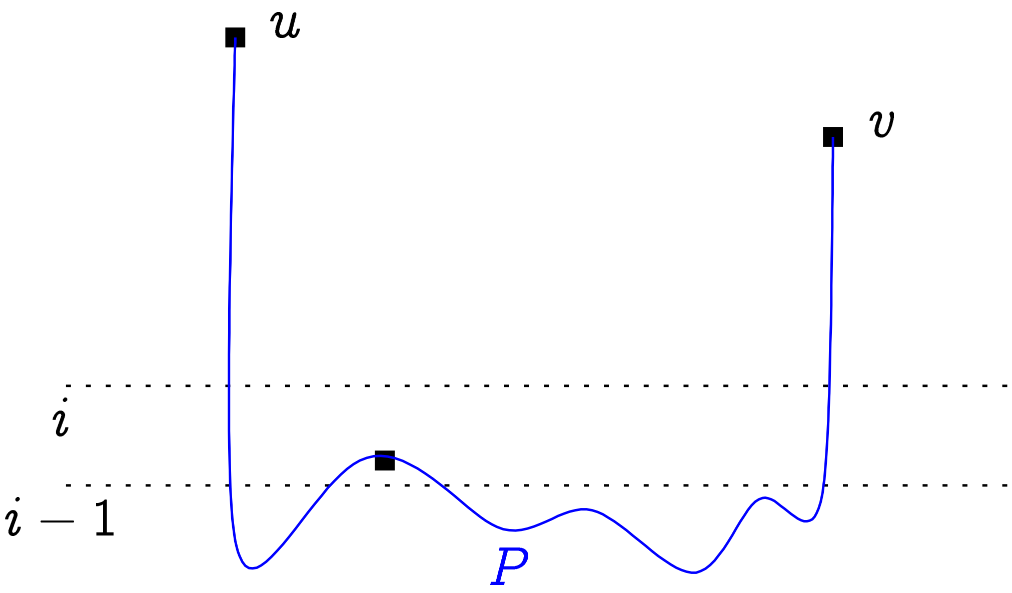
Defineerime veel $d^0(u, v) = \ell(u, v)$.

Millega võrdub $d^i(u,v)$. Tipp i võib antud tingimust rahuldavale lühimale teele kuuluda või mitte kuuluda.

Kui ei kuulu, siis $d^i(u,v) = d^{i-1}(u,v)$.



Kui kuulub, siis $d^i(u,v) = d^{i-1}(u,i) + d^{i-1}(i,v)$.



Tõepoolest, iga tipp kuulub lühimale ahelale ülimalt ühel korral. Seega on kõik teised tipud peale u, v ja i ahelal P hulgast $\{1, \ldots, i-1\}$.

Et liikuda võimalikult kiiresti tipust u tippu v, läbides seejuures ainult tippe $\{1, \ldots, i\}$ ja kindlasti läbides tippu i, tuleb

- liikuda võimalikult kiiresti tipust u tippu i, läbides seejuures ainult tippe $\{1, \ldots, i-1\}$,
- liikuda võimalikult kiiresti tipust i tippu v, läbides seejuures ainult tippe $\{1, \ldots, i-1\}$.

Kokkuvõttes:

$$d^i(u,v) = \min(d^{i-1}(u,v), d^{i-1}(u,i) + d^{i-1}(i,v)) \ .$$

Kõigi tippude vaheliste lühimate teede pikkuste leidmiseks leiame järjest $d^0, d^1, d^2, \ldots, d^n$.

Seda algoritmi nimetatakse *Floyd-Warshall*i algoritmiks.

for all $u, v \in V$ **do** $d[0][u,v] := \ell(u,v)$
for $i := 1$ **to** n **do**
    for all $u, v \in V$ **do**
        $d[i][u,v] := \min(d[i-1][u,v],$
            $d[i-1][u,i] + d[i-1][i,v])$
return $d[n]$

Keerukus — $O(|V|^3)$.

Me ei pea kogu massiivi d mälus hoidma. Igal hetkel läheb meil tarvis ainult $d[i]$-d ja $d[i-1]$-e. Seega piisab kahest $V \times V$ maatriksist.

Lühimad teed ise (s.t. — maatriksi Π) leiame nagu varem kirjeldatud, keerukusega $O(|V||E|)$.

Kui meil ei ole negatiivse pikkusega servi, siis saab kõigi tippude vaheliste kauguste leidmiseks kasutada Dijkstra algoritmi, rakendades seda iga tipu jaoks.

Tema keerukus $O(|V|^2 \log |V| + |V||E|)$ pole halvem kui Floyd-Warshalli algoritmi keerukus $O(|V|^3)$. Kui servi on vähe, s.t. $|E|$ on väiksem kui $\Theta(|V|^2)$, siis on Dijkstra algoritmi rakendamine iga tipu jaoks kiirem kui Floyd-Warshalli algoritm.

Järgnevas näitame, kuidas kasutada Dijkstra algoritmi ka juhul, kui graafis on negatiivse pikkusega servi. ($Johnson$i algoritm)

Tahame defineerida graafi servadel sellised uued pikkused $\hat{\ell}(v,w)$, et

- kui P on lühim tee u-st v-sse servade pikkuste ℓ järgi, siis on ta lühim tee u-st v-sse ka servade pikkuste $\hat{\ell}$ järgi;
- $\hat{\ell}(v,w) \geqslant 0$ iga $v, w \in V$ jaoks.

Kui oleme sellised pikkused defineerinud, siis kasutame Dijkstra algoritmi, kasutades pikkusi $\hat{\ell}$.

Olgu $h : V \longrightarrow \mathbb{R}$ suvaline funktsioon. Defineerime kõigi $u, v \in V$ jaoks $\hat{\ell}(u,v) = \ell(u,v) + h(u) - h(v)$.

Selliselt defineeritud $\hat{\ell}$ rahuldab esimest tingimust eelmiselt slaidilt. Tõepoolest, olgu $v_0 \to v_1 \to v_2 \cdots \to v_n$ mingi tee graafis G. Siis

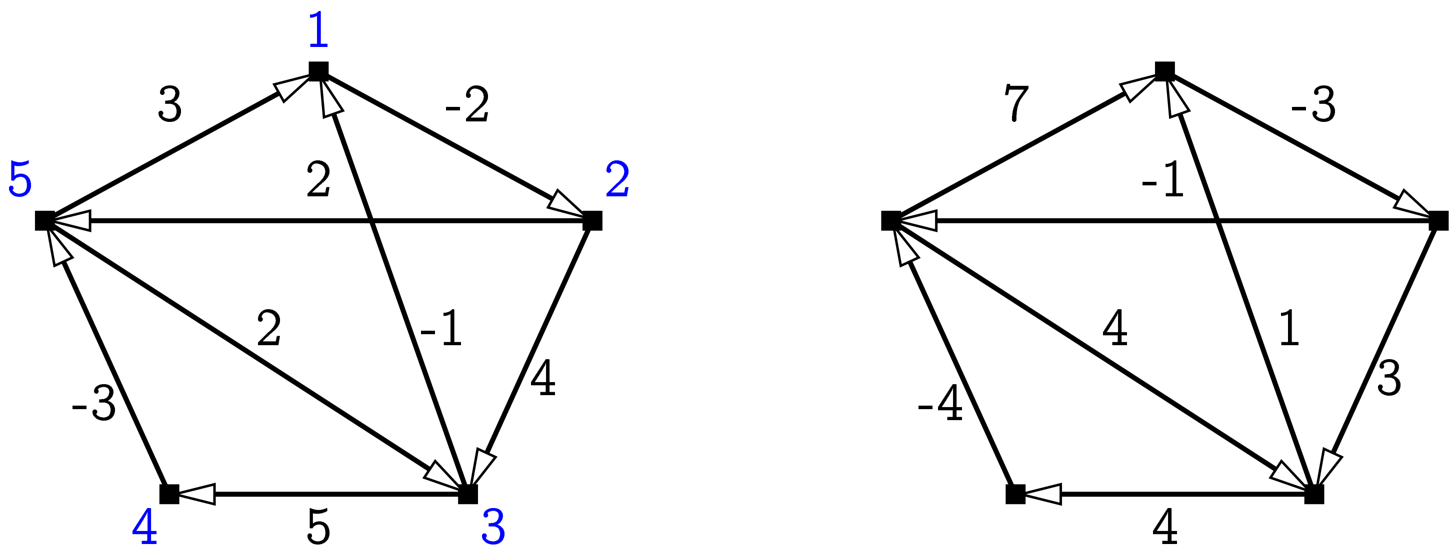
$$\hat{\ell}(P) := \sum_{i=1}^{n} \hat{\ell}(v_{i-1}, v_i) = \sum_{i=1}^{n} (\ell(v_{i-1}, v_i) + h(v_{i-1}) - h(v_i)) =$$

$$\sum_{i=1}^{n} \ell(v_{i-1}, v_i) + \sum_{i=0}^{n-1} h(v_i) - \sum_{i=1}^{n} h(v_i) =$$

$$\sum_{i=1}^{n} \ell(v_{i-1}, v_i) + h(v_0) - h(v_n) =: \ell(P) + h(v_0) - h(v_n) \quad .$$

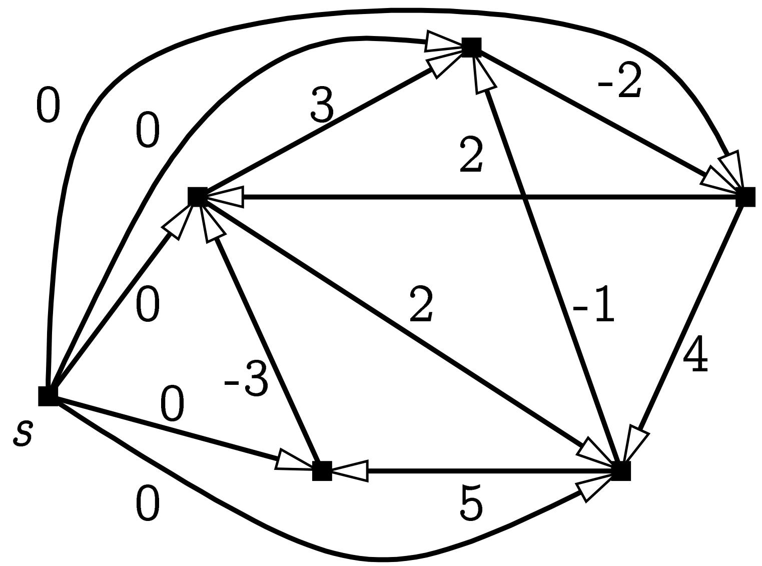
S.t. kõigi teede pikkus kahe fikseeritud tipu vahel muutub samapalju.

Näide ℓ-i muutmisest funktsiooni h abil.



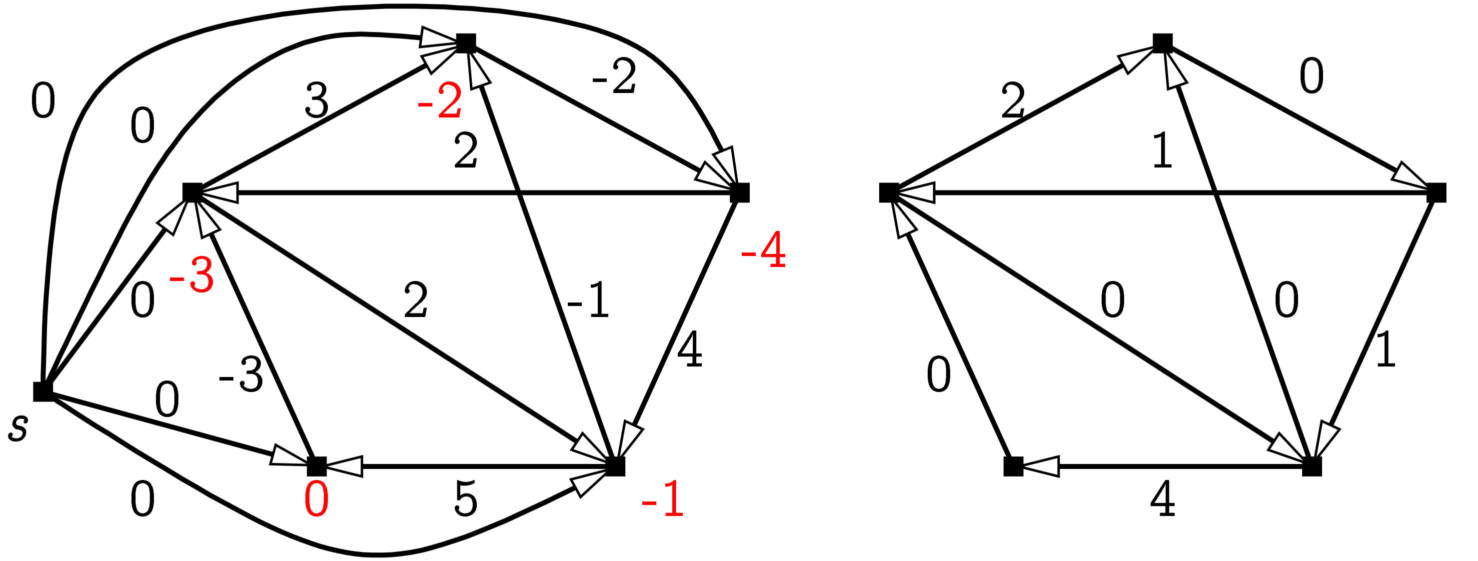
Me tahame h-i, mis kõik pikkused mittenegatiivseks muudaks.

Selleks lisame graafile ühe täiendava tipu s ja servad tipust s kõigisse teistesse tippudesse. Nende servade pikkuseks võtame 0.



Seejuures ei lisandu graafi tsükleid, sest ühestki tipust ei saa minna s-i. Samuti jäävad kõigi olemasolevate tsüklite pikkused samaks.

Ka uues graafis pole negatiivse pikkusega tsükleid. Bellman-Fordi algoritmi abil leiame tipu s kauguse kõigist teistest tippudest.



Võtame $h(v) := d(s,v)$.

Olles defineerinud $h(v) = d(s, v)$, on meil

$$\hat{\ell}(u, v) = \ell(u, v) + d(s, u) - d(s, v) \quad .$$

Meil on $d(s, v) \leqslant d(s, u) + \ell(u, v)$, seega $\hat{\ell}(u, v) \geqslant 0$.

Johnsoni algoritm:

- Lisa graafile täiendav tipp s, ühenda kõigi teiste tippudega 0-pikkusega servade abil.
- Leia Bellman-Fordi algortmiga s-i kaugus kõigist teistest tippudest.
- Esialgses graafis defineeri $\hat{\ell}(u,v) = \ell(u,v) + d(s,u) - d(s,v)$.
- Esialgse graafi iga tipu u jaoks leia Dijkstra algoritmiga tema kaugus ($\hat{\ell}$ järgi) kõigist teistest tippudest. Samuti leia lühimad teed ise.
- Leia lühimate teede pikkused ℓ järgi. Kui $\hat{\ell}$ järgi on u ja v kaugus $\hat{d}(u,v)$, siis $d(u,v) = \hat{d}(u,v) + d(s,v) - d(s,u)$.