

# Reiknirit Bellmans og Fords (1958 og 1956)

Bergur Snorrason

5. mars 2023

- ▶ Hvað gerum við ef við viljum nota reinkirit Dijkstras en það mega vera neikvæðar vigtir á leggjunum.
- ▶ Við getum þá notað reiknirit sem er kennt við Bellman og Ford.
- ▶ Við þurfum þó að fórna keyrslutíma.

- ▶ Þetta reiknirit er að vissu leiti einfaldara en reiknirit Dijkstras.
- ▶ Við notum kvika bestun og svörum spurningunni „Hver er stysta leiðin frá  $u$  til  $v$  sem fer að mestu í  $k$  hnúta?“.
- ▶ Hér táknar  $u$  upphafshnútin á meðan  $v$  og  $k$  eru frjálsar breytur.
- ▶ Látum þá  $f(v, k)$  tákna systa veg frá hnútnum  $u$  til hnútsins  $v$  sem fer ekki í fleiri en  $k$  hnúta.
- ▶ Til að einfalda skriftir þá skilgreinum við

$$E_u = \{v \in V : (u, v) \in E\}$$

og

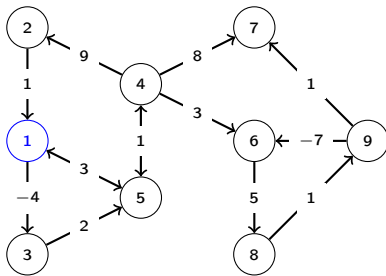
$$E^v = \{u \in V : (u, v) \in E\}.$$

- ▶ Við fáum að

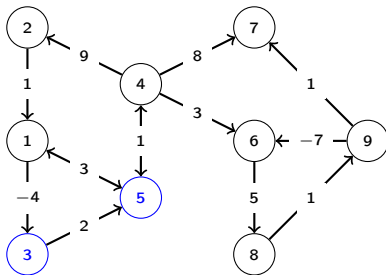
$$f(v, k) = \begin{cases} 0, & \text{ef } u = v \text{ og } k = 0 \\ \infty, & \text{ef } u \neq v \text{ og } k = 0 \\ \min(f(v, k-1), & \\ \min_{u \in E^v} w((u, v)) + f(u, k-1)), & \text{ef } u \neq v \text{ og } k = 0 \end{cases}$$

- ▶ Við munum leysa þetta með neðansækinni kvikri bestun.
- ▶ Gerum ráð fyrir að taflan sem við notum fyrir minnun hafi dálk sem svari til  $k$  breytunnar.
- ▶ Þá er hver staða aðeins háð stöðum í röðinni fyrir ofan sig.
- ▶ Við notum því aðeins síðustu línu fylkisins þegar við fyllum inn í töfluna.
- ▶ Því má geyma tvívíða fylkið sem einvitt fylki.

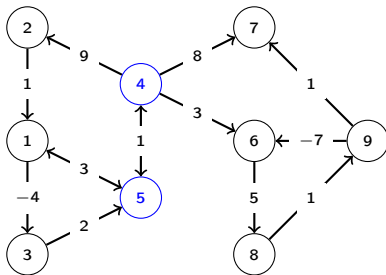
- ▶ Við erum ekki búin þegar við höfum reiknað öll gildin á  $f(v, k)$ .
- ▶ Hvað með neikvæðar rásir?
- ▶ Takið fyrst eftir að ef það er ekki neikvæð rás í netinu þá heimsækir systi vegur milli hnúta engan hnúta tvisvar.
- ▶ Einnig er ekki nóg að það sé neikvæð rás í netinu heldur þarf að vera hægt að komast í hana frá upphafshnútnum og svo má vera að það sé ekki hægt að komast frá rásinni í alla aðra hnúta.
- ▶ Við getum einfaldlega prófað að lengja vegina um  $V - 1$  hnúta í viðbót.
- ▶ Ef vegalengdin styttest einhverntíman þá er betra að heimsækja einhvern hnút oftar en einu sinni, sem þýðir að það sé neikvæð rás á leiðinni.



$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

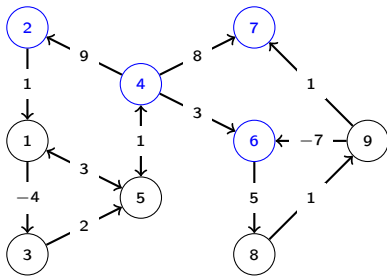


$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	$\infty$	-4	$\infty$	3	$\infty$	$\infty$	$\infty$	$\infty$

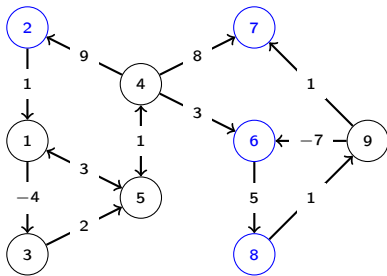


$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	$\infty$	-4	$\infty$	3	$\infty$	$\infty$	$\infty$	$\infty$
2	0	$\infty$	-4	4	-2	$\infty$	$\infty$	$\infty$	$\infty$

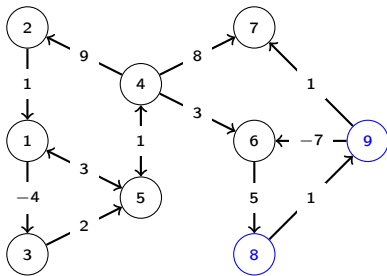




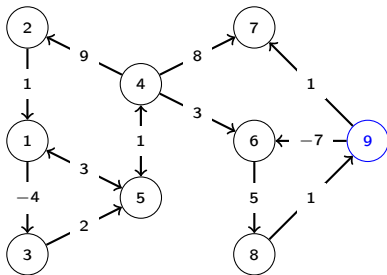
$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	$\infty$	-4	$\infty$	3	$\infty$	$\infty$	$\infty$	$\infty$
2	0	$\infty$	-4	4	-2	$\infty$	$\infty$	$\infty$	$\infty$
3	0	13	-4	-1	-2	7	12	$\infty$	$\infty$



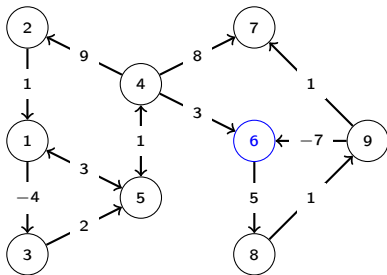
$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	$\infty$	-4	$\infty$	3	$\infty$	$\infty$	$\infty$	$\infty$
2	0	$\infty$	-4	4	-2	$\infty$	$\infty$	$\infty$	$\infty$
3	0	13	-4	-1	-2	7	12	$\infty$	$\infty$
4	0	8	-4	-1	-2	2	7	12	$\infty$



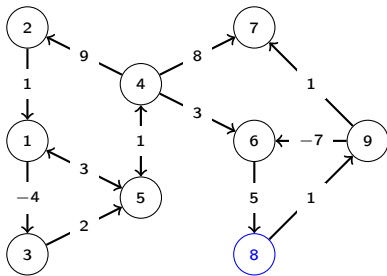
$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	$\infty$	-4	$\infty$	3	$\infty$	$\infty$	$\infty$	$\infty$
2	0	$\infty$	-4	4	-2	$\infty$	$\infty$	$\infty$	$\infty$
3	0	13	-4	-1	-2	7	12	$\infty$	$\infty$
4	0	8	-4	-1	-2	2	7	12	$\infty$
5	0	8	-4	-1	-2	2	7	7	13



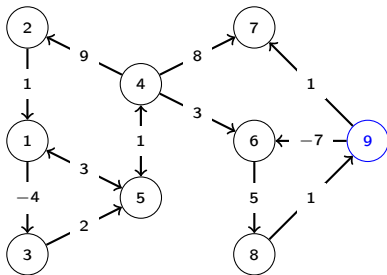
$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	$\infty$	-4	$\infty$	3	$\infty$	$\infty$	$\infty$	$\infty$
2	0	$\infty$	-4	4	-2	$\infty$	$\infty$	$\infty$	$\infty$
3	0	13	-4	-1	-2	7	12	$\infty$	$\infty$
4	0	8	-4	-1	-2	2	7	12	$\infty$
5	0	8	-4	-1	-2	2	7	7	13
6	0	8	-4	-1	-2	2	7	7	8



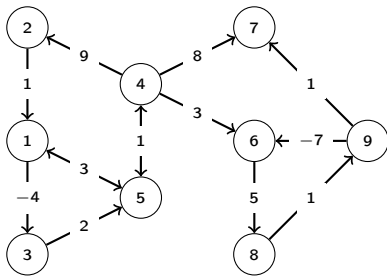
$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	$\infty$	-4	$\infty$	3	$\infty$	$\infty$	$\infty$	$\infty$
2	0	$\infty$	-4	4	-2	$\infty$	$\infty$	$\infty$	$\infty$
3	0	13	-4	-1	-2	7	12	$\infty$	$\infty$
4	0	8	-4	-1	-2	2	7	12	$\infty$
5	0	8	-4	-1	-2	2	7	7	13
6	0	8	-4	-1	-2	2	7	7	8
7	0	8	-4	-1	-2	1	7	7	8



$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	$\infty$	-4	$\infty$	3	$\infty$	$\infty$	$\infty$	$\infty$
2	0	$\infty$	-4	4	-2	$\infty$	$\infty$	$\infty$	$\infty$
3	0	13	-4	-1	-2	7	12	$\infty$	$\infty$
4	0	8	-4	-1	-2	2	7	12	$\infty$
5	0	8	-4	-1	-2	2	7	7	13
6	0	8	-4	-1	-2	2	7	7	8
7	0	8	-4	-1	-2	1	7	7	8
8	0	8	-4	-1	-2	1	7	6	8



$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	$\infty$	-4	$\infty$	3	$\infty$	$\infty$	$\infty$	$\infty$
2	0	$\infty$	-4	4	-2	$\infty$	$\infty$	$\infty$	$\infty$
3	0	13	-4	-1	-2	7	12	$\infty$	$\infty$
4	0	8	-4	-1	-2	2	7	12	$\infty$
5	0	8	-4	-1	-2	2	7	7	13
6	0	8	-4	-1	-2	2	7	7	8
7	0	8	-4	-1	-2	1	7	7	8
8	0	8	-4	-1	-2	1	7	6	8
9	0	8	-4	-1	-2	1	7	6	7



$k$	1	2	3	4	5	6	7	8	9
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	$\infty$	-4	$\infty$	3	$\infty$	$\infty$	$\infty$	$\infty$
2	0	$\infty$	-4	4	-2	$\infty$	$\infty$	$\infty$	$\infty$
3	0	13	-4	-1	-2	7	12	$\infty$	$\infty$
4	0	8	-4	-1	-2	2	7	12	$\infty$
5	0	8	-4	-1	-2	2	7	7	13
6	0	8	-4	-1	-2	2	7	7	8
7	0	8	-4	-1	-2	1	7	7	8
8	0	8	-4	-1	-2	1	7	6	8
9	0	8	-4	-1	-2	1	7	6	7



```

9 vi bellman_ford(vvii& g, int s)
10 {
11     int i, j, k, n = g.size(), x, w;
12     vi d(g.size(), INF);
13     d[s] = 0;
14     for (i = 0; i < n - 1; i++) for (j = 0; j < n; j++) if (d[j] != INF)
15         for (k = 0; k < g[j].size(); k++)
16             d[g[j][k].first] = min(d[g[j][k].first], d[j] + g[j][k].second);
17     for (i = 0; i < n - 1; i++) for (j = 0; j < n; j++) if (d[j] != INF)
18         for (k = 0; k < g[j].size(); k++)
19             {
20                 x = g[j][k].first, w = g[j][k].second;
21                 if (d[x] != -INF && d[j] + w < d[x]) d[x] = -INF;
22             }
23     return d;
24 }

```









- ▶ Sjáum að í fyrri hluta reikniritsins ýtrum við í gegnum alla leggi og allar hnúta  $(V - 1)$ -sinnum.
- ▶ Tímaflækjan á þeim hluta er því  $\mathcal{O}(E \cdot V)$ .
- ▶ Seinni hlutinn er svo að ítra yfir nákvæmlega það sama, svo tímaflækja þar er eins.
- ▶ Því fæst að reikniritið er í heildina  $\mathcal{O}(E \cdot V)$ .
- ▶ Þetta er töluvert verra en reiknirit Dijkstras (svipað og að fara úr  $\mathcal{O}(n \log n)$  í  $\mathcal{O}(n^2)$ ).

- ▶ Tökum eftir að það er ekki uppfært gildi í hnút  $v$  eftir legg  $(u, v)$  nema að  $u$  hafi verið uppfært í umferðinni áður.
- ▶ Í sýnidæminu svara þetta til bláu gildanna.
- ▶ Við getum notað þetta til að bæta keyrlsuhraðann, án þess þó að bæta tímaflækjuna.

```

9 vi bellman_ford(vvii& g, int s)
10 {
11     int i, j, k, n = g.size(), x, w, q[n];
12     vi d(n);
13     for (i = 0; i < n; i++) d[i] = i == s ? 0 : INF, q[i] = i == s ? 0 : -1;
14     for (i = 0; i < n - 1; i++) for (j = 0; j < n; j++) if (q[j] == i)
15         for (k = 0; k < g[j].size(); k++)
16             {
17                 x = g[j][k].first, w = g[j][k].second;
18                 if (d[j] + w < d[x]) d[x] = d[j] + w, q[x] = i + 1;
19             }
20     for (i = 0; i < n - 1; i++) for (j = 0; j < n; j++) if (q[j] == n + i - 1)
21         for (k = 0; k < g[j].size(); k++)
22             {
23                 x = g[j][k].first, w = g[j][k].second;
24                 if (d[x] != -INF && d[j] + w < d[x]) d[x] = -INF, q[x] = n + i;
25             }
26     return d;
27 }

```

DATE	JUDGEMENT	RUNTIME	LANGUAGE
12:01:53   	 Accepted	0.12 s	C++
11:50:38   	 Accepted	0.04 s	C++

