

Reiknirit Floyds og Warshalls (1962)

Bergur Snorrason

3. mars 2021

- ▶ Gerum ráð fyrir að við séum með vegið net.
- ▶ Hvað gerum við ef við viljum finna systa veg milli alla hnúta?
- ▶ Við getum beitt reinkiriti Dijkstra (eða reikniriti Bellmans og Fords) á alla hnúta.
- ▶ Það hefur tímaflækju $\mathcal{O}(V \cdot (V + E) \cdot \log E)$ (eða $\mathcal{O}(E \cdot V^2)$).
- ▶ Þetta má þó bæta með kvikri bestun.

- ▶ Gerum ráð fyrir að $V = \{1, 2, \dots, n\}$.
- ▶ Látum $f(u, v, k)$ tákna stysta veg milli hnútanna u og v sem heimsækir einhvern hnútanna $1, 2, \dots, k$ á milli.
- ▶ Fyrir fast k gildir að stysti vegur milli u og v undir slíkum skorðum fer annað hvort gegnum hnút k eða ekki.
- ▶ Þetta gefur okkur tvö tilfelli.
- ▶ Við fáum

$$f(u, v, k) = \begin{cases} d_{uv}, & \text{ef } k = 0. \\ \min(f(u, v, k-1), f(u, k, k-1) + f(k, v, k-1)), & \text{annars} \end{cases}$$

þar sem d_{uv} táknar fjarlægð frá hnút u til hnúts v í netinu.

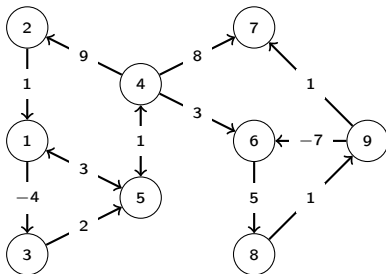
- ▶ Þetta má að sjálfsgöðu útfæra ofansækið en venjan er að gera það neðansækið.

```

1 vvi floydwarshall(vvii& g)
2 {
3     int i, j, k, n = g.size();
4     vvi dp(n, vi(n, INF));
5     rep(i, n) dp[i][i] = 0;
6     rep(i, n) rep(j, g[i].size())
7         dp[i][g[i][j].first] = min(g[i][j].second, dp[i][g[i][j].first]);
8     rep(k, n) rep(i, n) rep(j, n)
9     {
10         if(dp[i][k] == INF || dp[k][j] == INF) continue;
11         dp[i][j] = min(dp[i][j], dp[i][k] + dp[k][j]);
12     }
13     rep(k, n) rep(i, n) rep(j, n)
14     {
15         if(dp[i][k] == INF || dp[k][j] == INF || dp[i][j] == INF) continue;
16         if(dp[i][k] + dp[k][j] < dp[i][j]) dp[i][j] = -INF;
17     }
18     return dp;
19 }

```

- ▶ Þar sem við erum með þrívítt stöðurúm, hver vídd hefur stærð $|V|$ og reikna má fallgildi f í $\mathcal{O}(1)$ tíma fæst að reinkiritið hefur tímaflækjuna $\mathcal{O}(V^3)$.



	1	2	3	4	5	6	7	8	9
1	0	∞	-4	3	∞	∞	∞	∞	∞
2	1	0	∞	∞	∞	∞	∞	∞	∞
3	∞	∞	0	∞	2	∞	∞	∞	∞
4	∞	9	∞	0	1	3	8	∞	∞
5	3	∞	∞	1	0	∞	∞	∞	∞
6	∞	∞	∞	∞	∞	0	∞	5	∞
7	∞	∞	∞	∞	∞	∞	0	∞	∞
8	∞	∞	∞	∞	∞	∞	∞	0	1
9	∞	∞	∞	∞	∞	-7	1	∞	0
	1	2	3	4	5	6	7	8	9
1	0	∞	-4	3	∞	∞	∞	∞	∞

