Dijkstra

```
Function Dijkstra(G, zdroj v)
    stav[*] ← N; dist[*] ← ∞; P[*] ← ∅
    stav[v] ← O; dist[v] ← 0
    halda.Insert(v)
    while ∃ otevřené vrcholy do
        u ← halda.ExtractMin()
        for w soused u do
            if dist[w] > dist[u] + d(uw) then
                dist[w] ← dist[u] + d(uw)
                if stav[w] = N then
                    halda.Insert(w)
                end
                else
                    halda.Decrease(w)
                end
                stav[w] ← O
            end
        end
        stav[u] ← Z
    end
    return pole dist
end
```

Korektnost

- **Nezápornost -** prodloužením sledu o hranu se jeho hodnota (délka) nemůže snížít - vlastnost úlohy
- **Monotonie** - invariant : v každém okamžiku mají uzařené vrcholy nižší (nebo rovné) aktuální vzdálenosti než otevřené vrcholy
- Dijkstra uzavírá vrcholy nejvýše jednou, a to v pořadí dle skutečné vzdálenosti

Složitost Dijkstrova algoritmu
Verze s decrease:

- čas: $\mathcal{O}(n * Insert + n * Extract + m * Decrease)$
- prostor: $\mathcal{O}(n)$

Verze bez decrease:

- čas: $\mathcal{O}(m * Insert + m * Extract) = \omega(m \log n)$
- prostor: $\mathcal{O}(m)$

Table 1: Složitost dle hald

Typ haldy	**Insert**	**Extract**	**Decrease**	**Dijkstra**
Pole/Seznam	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n^2)$
Strom/Bináni	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	$\mathcal{O}((n+m)\log n)$
Fibonacci	$\mathcal{O}_a(1)$	$\mathcal{O}_a(\log n)$	$\mathcal{O}_a(1)$	$\mathcal{O}(m + n\log n)$

Relaxační Algoritmy

Function *RelaxAlg(G, zdroj v)*
  stav[*] $\leftarrow$ N; vzdalenost[*] $\leftarrow \infty$; P[*] $\leftarrow \emptyset$
  stav[v] $\leftarrow$ O; dist[v] $\leftarrow 0$
  while $\exists$ *otevřené vrcholy* **do**
    $u \leftarrow$ vyber otevřený vrchol
    Relaxuj u (zlepši odhady sousedů, a otevři zlepšené)
  end
  return pole dist
end

Strategie relaxace

- Od nejmenší vzdálenosti, jako Dijkstra $\rightarrow \mathcal{O}(exp(n))$ worst case
- Zásobník, jako DFS $\rightarrow \mathcal{O}(exp(n))$ worst case
- Fronta $\rightarrow \mathcal{O}(nm)$ worst case (Bellman-Ford)

Korektnost Bellman-Forda

- Koncept: Bellman-Ford prochází vrcholy ve fázích, ve fázi $i+1$ jsou relaxovány vrcholy otevřené ve fázi i
- Po fázi i byly všechny vrcholy ohodnoceny nejkrajšími cestami s nejvýše i hranami (resp. byly prozkoumány všechny sledy délek nejvýše i)

Floyd-Warshall

```
Function FloydWarshall(D^0 matice délek)
    for fáze k = 1, ..., n do
        for vrchol u = 1, ..., n do
            for vrchol v = 1, ..., n do
                D^k_{u,v} ← min(D^{k-1}_{u,v}, D^{k-1}_{u,k} + D^{k-1}_{k,v})
            end
        end
    end
    return matice vzdáleností D^n
end
```

Korektnost Floyd-Washalla

- Pro každou nejkratší cestu $P_{u \to v}$ z u do v (s více než jednou hranou) existuje nejvyšší index vnitřního vrcholu k
- $P_{u \to v}$ lze vyjádřit jako konkatenaci nejkratších cest $u \to k$ a $k \to v$ používajících vnitřní vrcholy pouze indexů menších než k
- Ve fázi k jsou zváženy všechny cesty používající vnitřní indexy nejvýše k