## Nejkratší cesty: Průzkumnický algoritmus

*Hledej*$(v_0)$ :

1. $Z(*) \leftarrow \mathbf{N}$, $Z(v_0) = \mathbf{O}$ značky: **N**eviděn, **O**tevřen, **U**zavřen
2. $D(*) \leftarrow \infty$, $D(v_0) = 0$ odhady vzdáleností
3. $P(*) \leftarrow \mathbf{?}$ předchůdci, **?**=nedefinováno
4. Dokud existuje vrchol $u$ takový, že $Z(u) = \mathbf{O}$:
5. Prozkoumáme vrchol $u$, čili:
6. $Z(u) \leftarrow \mathbf{U}$
7. Pro všechny vrcholy $v$, do kterých vede hrana z $u$:
8. Je-li $D(u) + \ell(u,v) < D(v)$:
9. $D(v) \leftarrow D(u) + \ell(u,v)$
10. $P(v) \leftarrow u$
11. $Z(v) \leftarrow \mathbf{O}$

**Věta:** Pokud se alg. zastaví, pak $\forall v\ D(v) = d(v_0, v)$ a graf $C = (V_C, E_C)$, $V_C = \{v \in V \mid Z(v) \neq \mathbf{N}\}, E_C = \{(v, P(v)) \mid v \in V_C \wedge P(v) \neq \mathbf{?}\}$, je *strom nejkratších cest.*

# Nejkratší cesty: Dijkstrův algoritmus

*Dijkstra*$(v_0)$ :

1. Inicializace jako u Průzkumnického algoritmu
2. Dokud existuje vrchol $u$ takový, že $Z(u) = \mathbf{O}$:
3. Zvolíme $u$ tak, aby $Z(u) = \mathbf{O}$ a $D(u)$ bylo minimální.
4. Prozkoumáme vrchol $u$.

**Lemma:** Algoritmus jednou uzavřený vrchol znovu neotevře.

**Důsledek:** Algoritmus se zastaví po nejvýše $n$ průchodech a vydá vzdálenosti z $v_0$ a strom nejkratších cest.

## Dijkstrův algoritmus: Jádro pudla

*Dijkstra*$(v_0)$ :

1. $M \leftarrow V$ množina neviděných a otevřených vrcholů
2. $D(*) \leftarrow \infty$, $D(v_0) = 0$ Insert
3. Dokud $M \neq \emptyset$:
4. Zvolíme $u \in M$, jehož $D(u)$ je minimální. DeleteMin
5. $M \leftarrow M \setminus \{u\}$
6. Je-li $D(u) = \infty$, skončíme.
7. Pro všechny vrcholy $v$, do kterých vede hrana z $u$:
8. $D(v) \leftarrow \min(D(v), D(u) + \ell(u, v))$ Decrease
   (nutné pouze pokud $v \in M$)

**Pozorování:** Potřebujeme datovou strukturu pro uchování všech $D(\cdot)$, která bude umět operace *Insert*, *DeleteMin* a *Decrease*.

Časová složitost pak bude $\mathcal{O}(n \cdot T_{Ins} + n \cdot T_{DelMin} + m \cdot T_{Dec})$.