

Dijkstrin algoritam  
na steroidima:

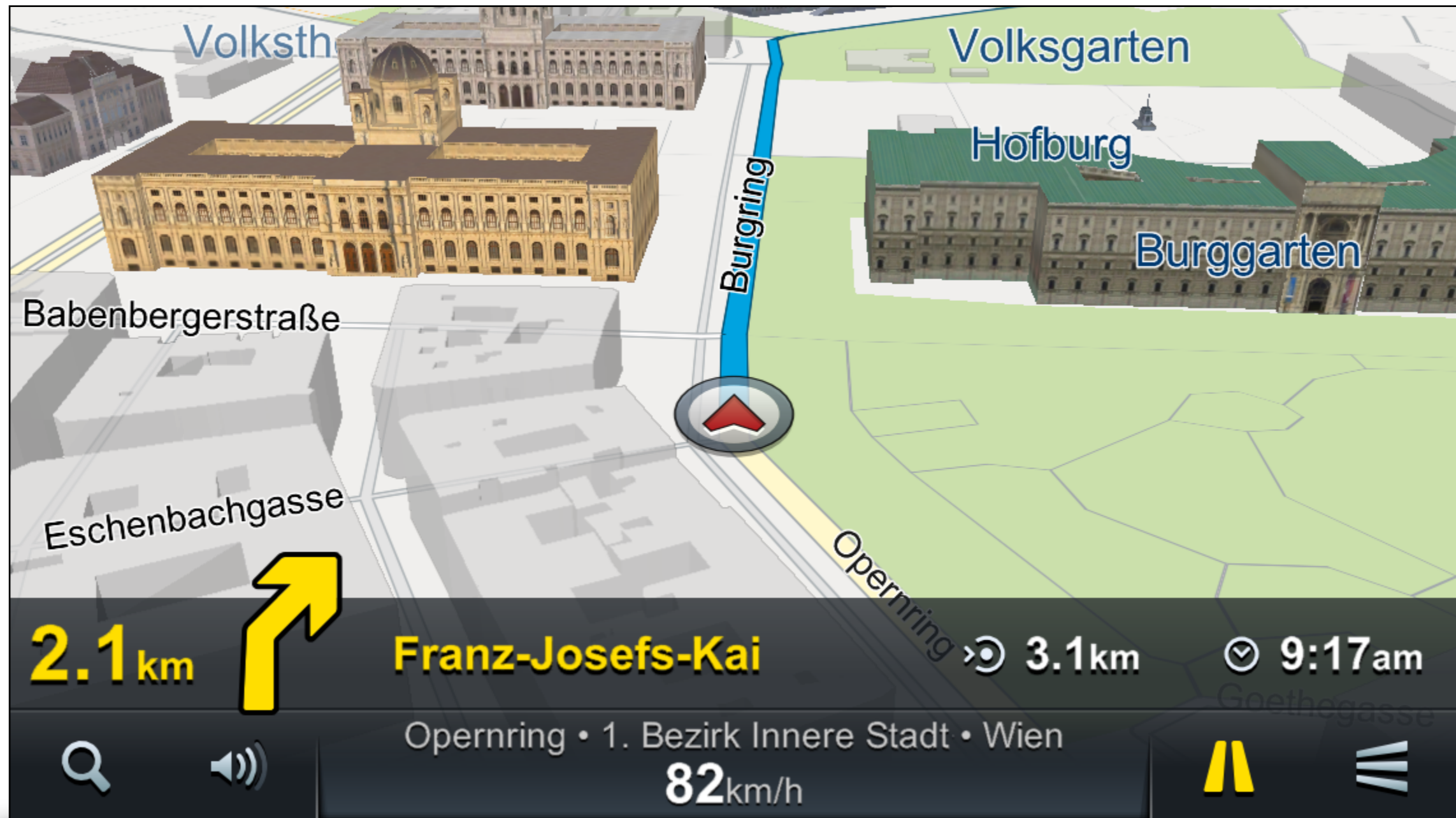
Sažete hijerarhije

Mireo

Software development and innovation company



# PROFESIONALNI CESTOVNI GPS NAVIGACIJSKI SUSTAVI



## SVJETSKO TRŽIŠTE

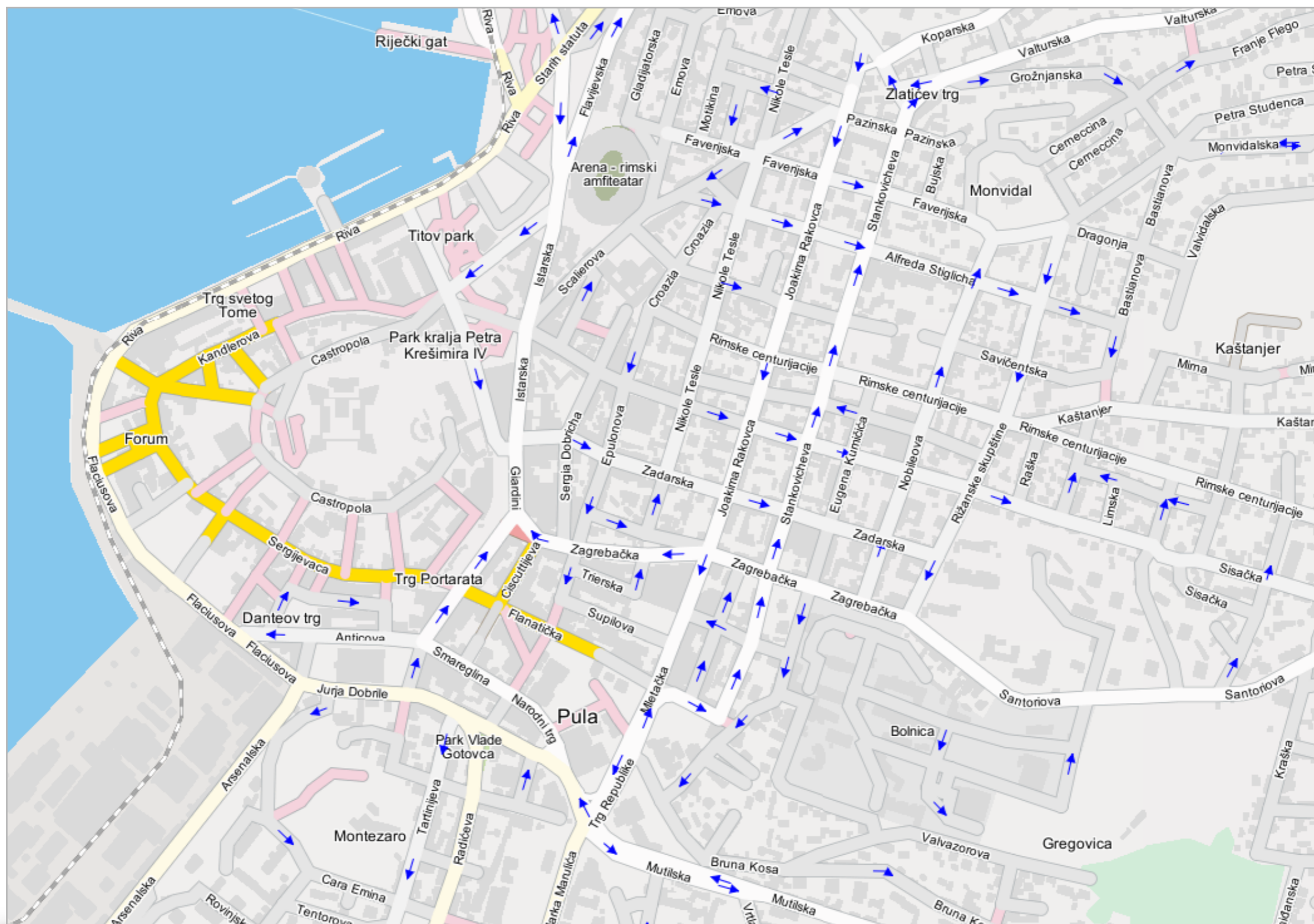
- U svijetu se godišnje proda više od 50 milijuna navigacijskih uređaja
- Platforme: Embedded Linux, Windows CE, iOS, Android, Blackberry, QNX

Mireo je jedan od vodećih nezavisnih proizvođača softvera za GPS navigacijske uređaje u svijetu



# RAZINA KARTOGRAFSKIH DETALJA - PULA

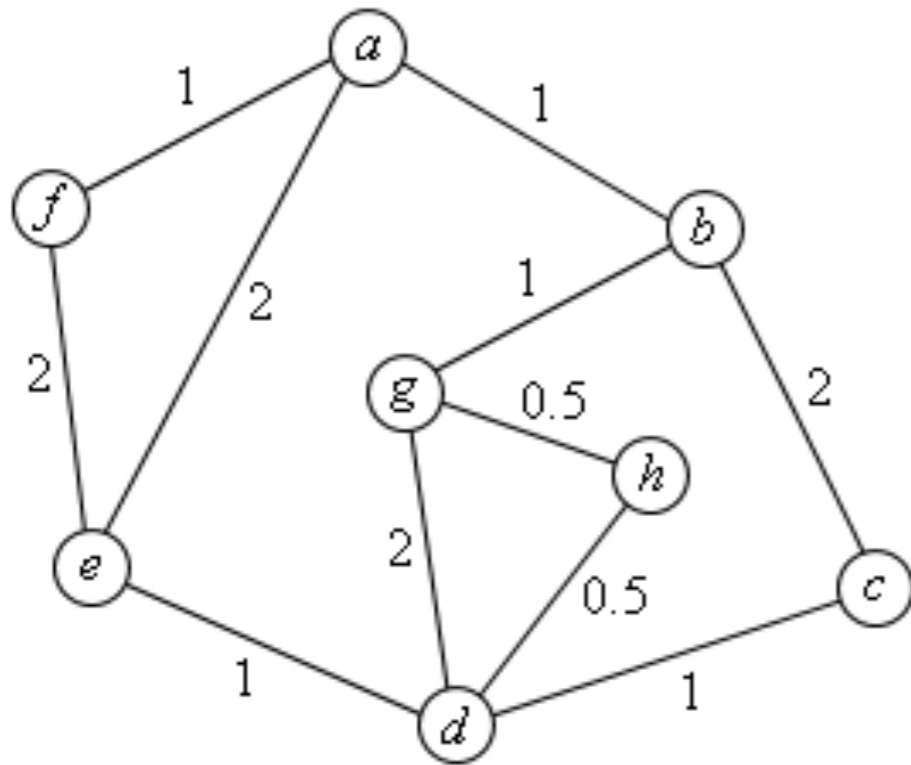
Mireo



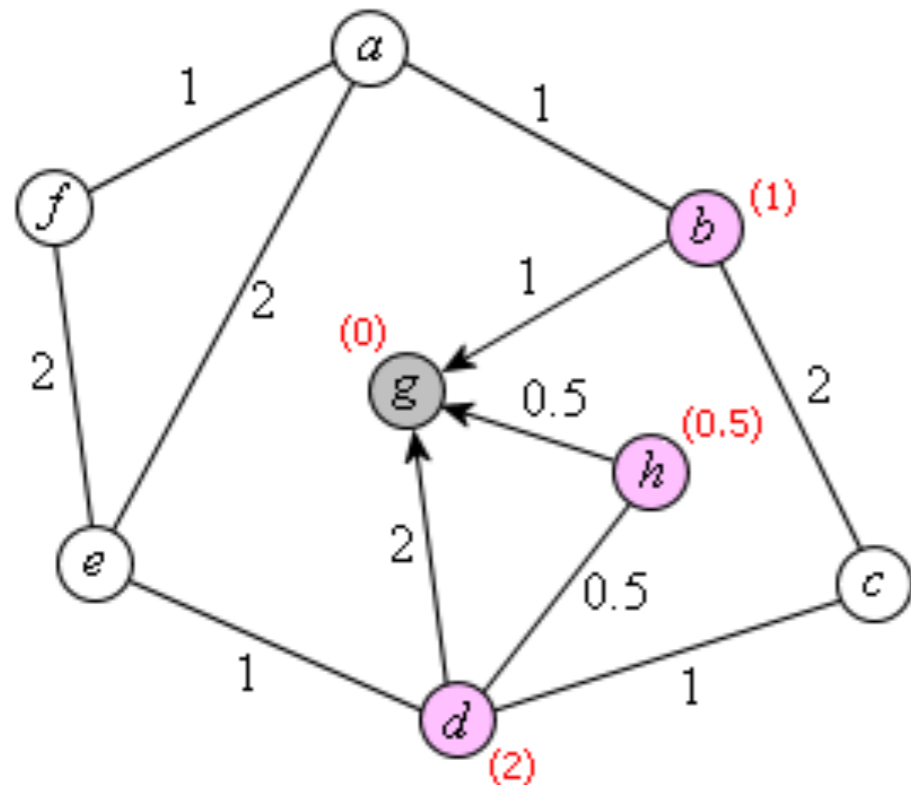
# CESTOVNI PUT ZAGREB-LONDON

Mireo





- vrhovi (čvorovi) grafa su križanja
- bridovi (grane) su cestovni segmenti među križanjima



- složenost Dijkstrinog algoritma je  $O(m + n \cdot \log(n))$

$m$  – broj grana

$n$  – broj čvorova

## KRIŽANJA I SEGMENTI U STVARNIM KARTAMA

	# križanja	#segmenata
Zapadna Europa	$18 \cdot 10^6$	$42.2 \cdot 10^6$
Europa	$40.4 \cdot 10^6$	$85.7 \cdot 10^6$
USA	$23.9 \cdot 10^6$	$57.7 \cdot 10^6$

Složenosti Dijkstrinog algoritma za Europu: ~  $O(900.000.000)$

## TRIKOVI

- Traži paralelno u okolini početne i završne točke (“dvosmjerno”) unutar malog radijusa (npr. 20 km). U traženju koristi sve ceste.
- Nastavi pretragu u većem radijusu (npr. 100 km), ali uzimajući u obzir samo državne ceste i autoceste.
- Nastavi pretragu uz korištenje isključivo autocesta.

## NEDOSTACI

- Veliki problem s netočno izračunatim putevima nastalim zbog heuristika
- Još uvijek preveliki memorijski i računski zahtjevi

## PRIMJER

Prosječno vrijeme izračunavanja najkraćeg puta od Karlsruhe-a do jednog od 13 glavnih gradova zapadnoeuropskih država je **59 sekundi** na komercijalnom uređaju.



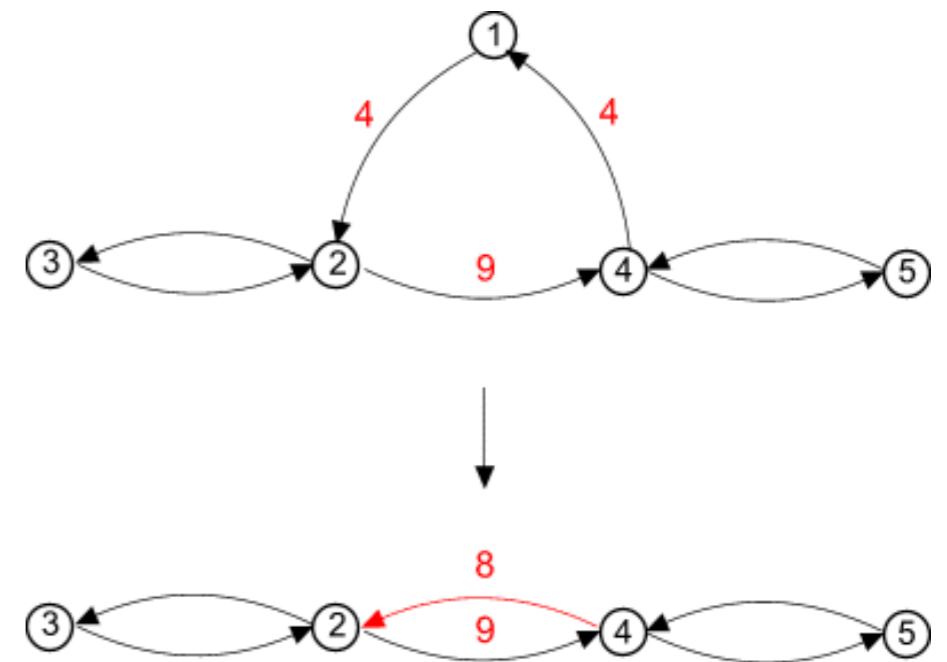
## CILJEVI

- Računanje puteva mora biti matematički egzaktno
- Optimizacije i kreiranje hijerarhijskih nivoa ne smiju ovisiti o klasifikaciji prometnica
- Algoritam mora raditi u uvjetima mobilnih uređaja (ARM procesor, limitirana radna i vanjska memorija)
- Računanje proizvoljnog puta u Europi mora trajati **kraće od 1 sekunde**



## KONSTRUKCIJA

- Određivanje linearnog uređaja na skupu vrhova (enumeriranje vrhova  $1..n$ ) obzirom na “važnost” križanja.
- Izbacivanje jednog po jednog vrha u redoslijedu od najmanjeg prema najvećem (obzirom na uređaj) uz uvjet da su najkraći putevi u preostalom grafu prekrivaču sačuvani.
- Konstrukcija grafa za traženje iz grafa prekrivača korištenjem samo “uzlaznih” grana.



Sažimanje vrha 1

## UZLAZNI GRAF

$G_{\uparrow} := (V, E_{\uparrow})$  gdje je  $E_{\uparrow} := \{(u, v) \in E^* : u < v\}$

## SILAZNI GRAF

$G_{\downarrow} := (V, E_{\downarrow})$  gdje je  $E_{\downarrow} := \{(u, v) \in E^* : u > v\}$

- Za pronalazak najkraćeg  $s$ - $t$  puta koristimo dvosmjerni modificirani Dijkstrin algoritam.
- U smjeru od početnog vrha  $s$  koristimo samo uzlazni graf  $G_{\uparrow}$ .
- U smjeru od ciljnog vrha  $t$  koristimo samo silazni graf  $G_{\downarrow}$ .
- Ako i samo ako postoji najkraći  $s$ - $t$  put u  $G$  obje će se pretrage susresti u vrhu  $v$  čiji je redni broj veći od svih drugih rednih brojeva vrhova na  $s$ - $t$  putu.

## PROSJEČNO VRIJEME RAČUNANJA SLUČAJNO ODABRANOG PUTA (UREĐAJ S ARM PROCESOROM)

### Vrijeme računanja

**Zapadna Europa**

**72.4 ms**

**Europa**

**84.2 ms**

**USA**

**56.5 ms**

Ubrzanje: ~ **3000 puta!**