



LUNDS  
UNIVERSITET  
Lunds Tekniska Högskola

# Inlämningsuppgift:

Kortaste vägen i en graf:  
Dijkstras algoritm

Javier Poremski (GDA10APO)

Programmering i C++  
EDAF30  
2017-12-18

## Design

Inlämningsuppgiften består totalt av fyra klasser: *Edge*, *Node*, *NodeSet*, *Graph*; samt två fria funktioner innehållandes Dijkstras algoritm.

### Edge

Klassen beskriver algoritmens bågkostnader i en viktad och riktad graf och håller reda på bågens noddestination.

### Node

Klassen hanterar algoritmens nod och innehåller dess namn, värde samt nodens nodförälder för att möjliggöra bakåtvandring i nodstegen.

### NodeSet

Klassen håller reda på nodmängder genom en vektor av typen *Node*. Därtill innehåller den en funktion som returnerar den nod som har lägsta värde där den returnerade noden tas bort ur mängden i *NodeSet*.

### Graph

Klassen behandlar en godtycklig graf där noderna är sparade i en vektor av typen *Node*.

Klassen tar in *istream*-indata i konstruktorn som möjliggör att en graf kan skapas från en textfil. Textfilen framträder enligt följande format per rad: `a: b c` där *a* är initialnod, *b* är kostnaden och *c* är destinationsnod.

### Dijkstras algoritm

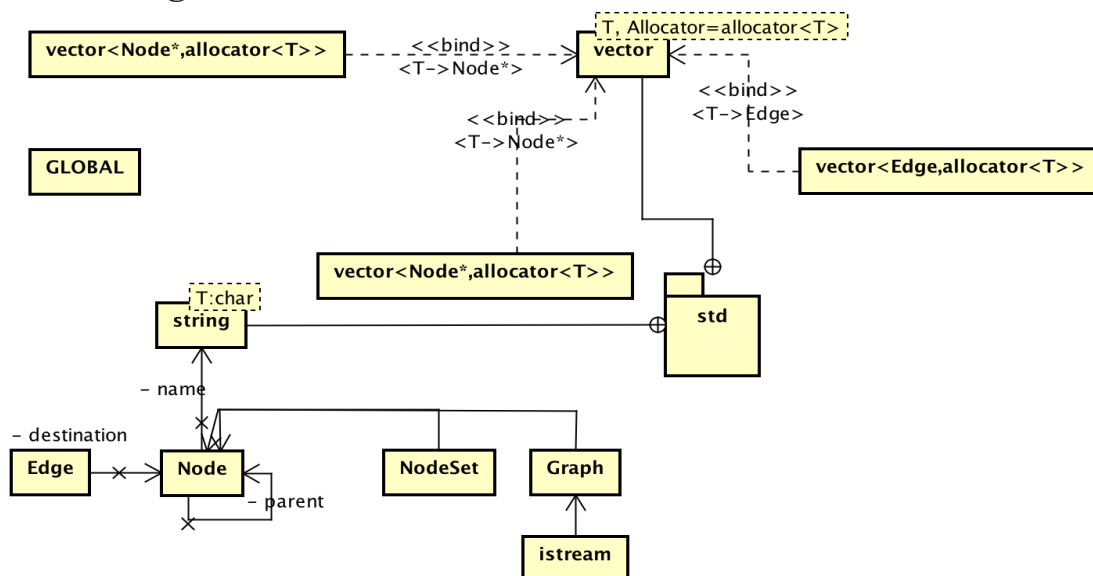
Algoritmen återfinns i filen *dijkstras.cc* och uppenbarar sig i två fria funktioner:

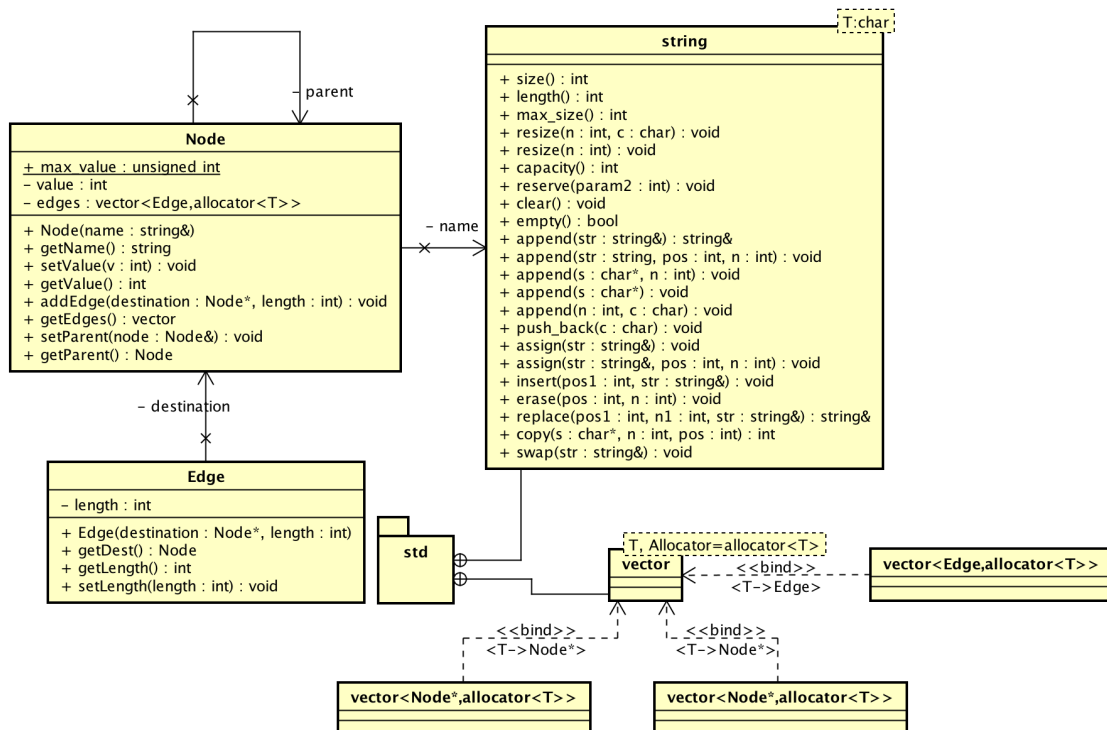
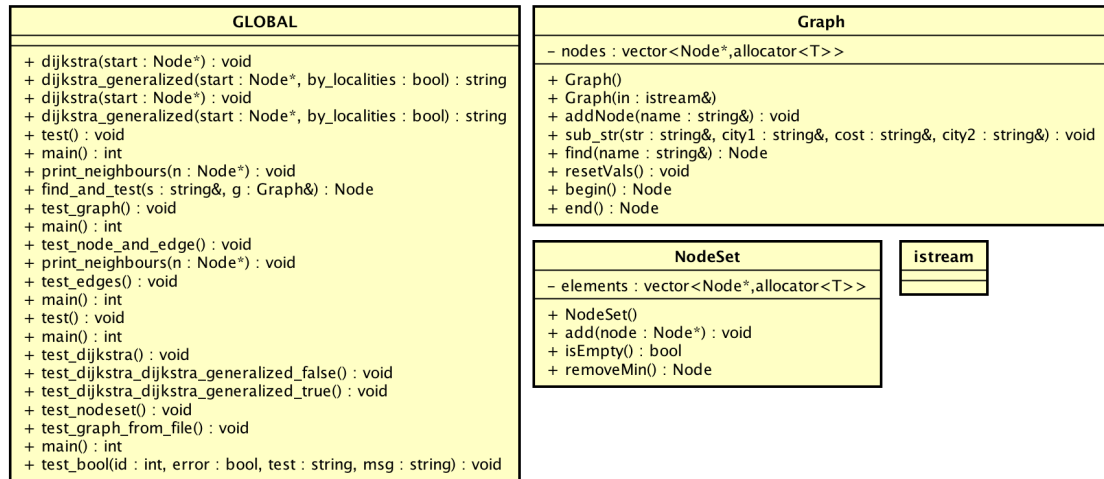
```
void dijkstra(Node* start);
```

```
std::string dijkstra_generalized(Node* start, bool by_localities);
```

Den förstnämnda funktionen motsvarar ursprungsfunktionen utifrån uppgiftbeskrivningen medan den sistnämnda funktionen är den generaliserade funktionen.

## Klassdiagram





## Användarinstruktion

Uppgiftens källkod är komprimerad i en Zip-fil varefter extrahering krävs kompilering genom att köra `make` i konsolen inne i den extraherade mappen. Efter att filerna har kompilerats möjliggör nedankommande kommandon en effektivare körning av testfilerna:

- `make run_all` kör samtliga testfiler, både de nerladdningsbara samt egenskrivan.
- `make run_pre` kör endast de nerladdningsbara testfilerna med förinställt makro till INFO.
- `make run_own` kör endast den egenskrivna testfilen.
- `make clean` tar bort samtliga kompilerade program som skapats i samband med körningen av `make`.

- `make distclean` kör "make clean" samt tar bort samtliga referenser till beroenden som har skapats vid kompileringen i samband med körningen av `make`.

## Testfiler

De nerladdningsbara testfilerna består av förledsnamnet `test_` undantages den egenskrivna testfilen vid namn `test_own.cc`. Den egenskrivna testfilen innehåller totalt fyra funktionstester: test av den ursprungliga Dijkstras algoritm, test av den generaliserade algoritmen, test av NodeSet (som hänger ihop med Node och Edge) samt test av Graph genom textfilsinläsning.