Hashmaps

```
H = new Heap[Vertex](Cost: HashMap[Vertex,Double]) // datastructure needs access to cost fu
H.insert(v)
V = H.removeMin();
H.decreaseKey(v);
H.contains(v);
    // implemented as array
    // array can resize itself - so infinitely large Heap

cost(v)
cost.contains(v)
cost.update(v,c)
----
if v.d > u.d + v(u,v) then
    v.d := u.d + w(u,v)
----
if (!cost.contains(v) || cost(v) > cost(u) + weight(u,v) {
    cost.update
}
----
-- elevationGain
weight(u,v) = (v.location-u.location).length + dz^2*1000
    -- subtracting point3d gives vector3d
    -- dz = positive if elevation gain; 0 otherwise (not sure if I want it)
```

You can mix math inline $x^2 + y^2 = z^2$ or in display mode: - Use a hashtable

$$\frac{c}{d} + x_2^2$$

adsf