Graph labeling

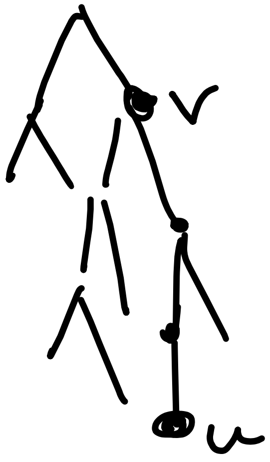
Algo21

Goal of labels: finding neighbors and distance

node → label

Representation:
matrix –
adj. list –
Instead:
labels

- Are two nodes adjacent?
- What is the distance between two nodes?
- …?



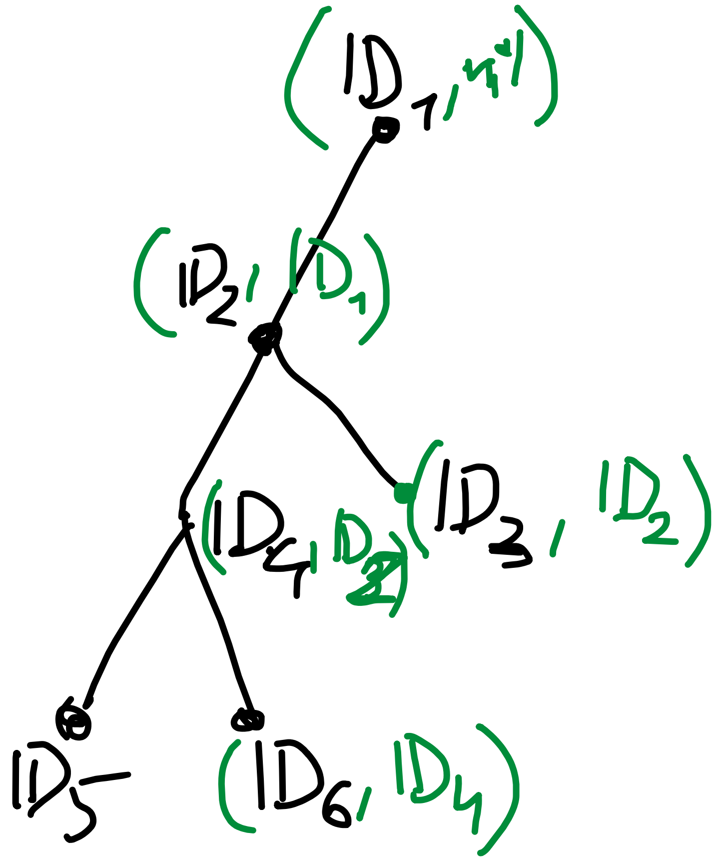
Answer based on labels of the nodes only:

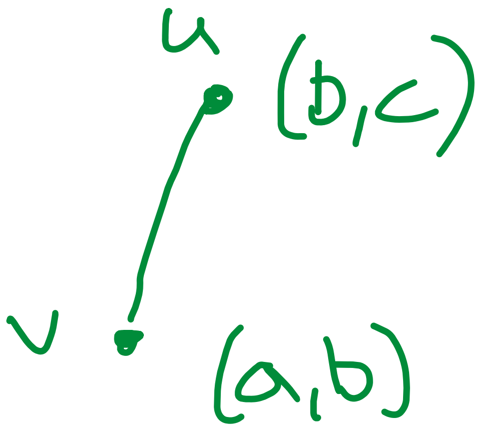
F(label(u), label(v))

Adjacency in a tree

2log(n) bit labels:

- log(n) bit ID for each node
- label= (node ID, parent's ID)





Labeling schemes for general graphs

The scheme should be applicable to any graph

Is it possible to get an efficient solution?

Lower bound:

Labels need to have the length $\Omega(n)$

simple

n-nodes

$v_1, \dots \quad v_n$

label v_i :

$(b_1, b_2, \dots \quad b_n)$

$b_j = 1 \iff (v_j, v_i) \in E$

Labeling schemes for general graphs

The scheme should be applicable to any graph

Is it possible to get an efficient solution?

Lower bound:

Labels need to have the length Ω(n)

$s \geq \frac{n}{2}$

simple

n-nodes

$v_1, \dots \quad v_n$

label v_i :

$(b_1, b_2, \dots \quad b_n)$

$b_j = 1 \Leftrightarrow (v_j, v_i) \in E$

Arguments for lower bound

1) Labeling for non-isomorphic graphs must be different

- If label length is s, then there are 2^s different labels
- Each labeling takes a subset of labels
- Two subsets cannot be taken for non-isomorphic graphs

$$\#\text{graphs} \leq \left(\!\!\binom{2^s}{n}\!\!\right) = \binom{2^s+n-1}{n}$$

- number of nonisomorphic graphs

$$\#\text{graphs} \geq 2^{\binom{n}{2}} / n!$$

$$G_1 \xrightarrow{\text{iso}} G_2$$

$$v_1, v_2, \dots, v_n \qquad v_1, v_2, \dots, v_n$$

$$(v_j, v_i) \in E_1 \Leftrightarrow f(v_j, v_i) = 1 \Leftrightarrow (v_j, v_i) \in E_2$$

Arguments for lower bound

1) Labeling for non-isomorphic graphs must be different

2) If label length is s, then there are 2^s different labels

3) Each labeling takes a subset of labels

n out of 2^s labels

- Two subsets cannot be taken for non-isomorphic graphs

$$\#\text{graphs} \le \left(\!\!\binom{2^s}{n}\!\!\right) = \binom{2^s+n-1}{n} \approx \frac{(2^s)^n}{n!}$$

$$(2^s)^n \geq 2^{n^2/2}$$

- number of nonisomorphic graphs

$$\#\text{graphs} \geq 2^{\binom{n}{2}} / n! \approx \frac{2^{n^2/2}}{n!}$$

permutations

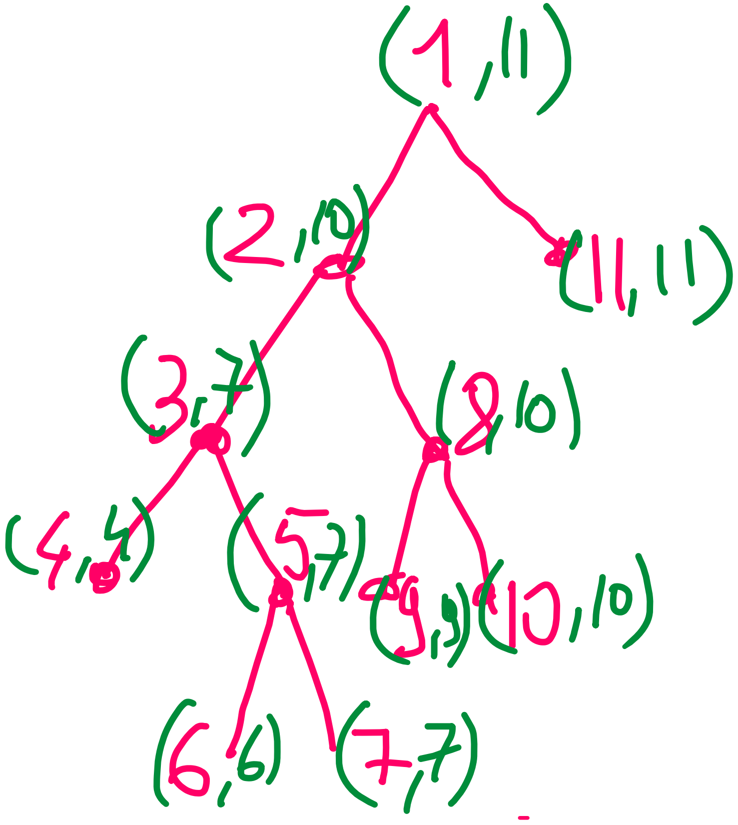
$$s \geq n/2$$

Ancestry question in a tree

labelling based on DFS

for each node – index in DFS search

label= (node index, highest index in a subtree)



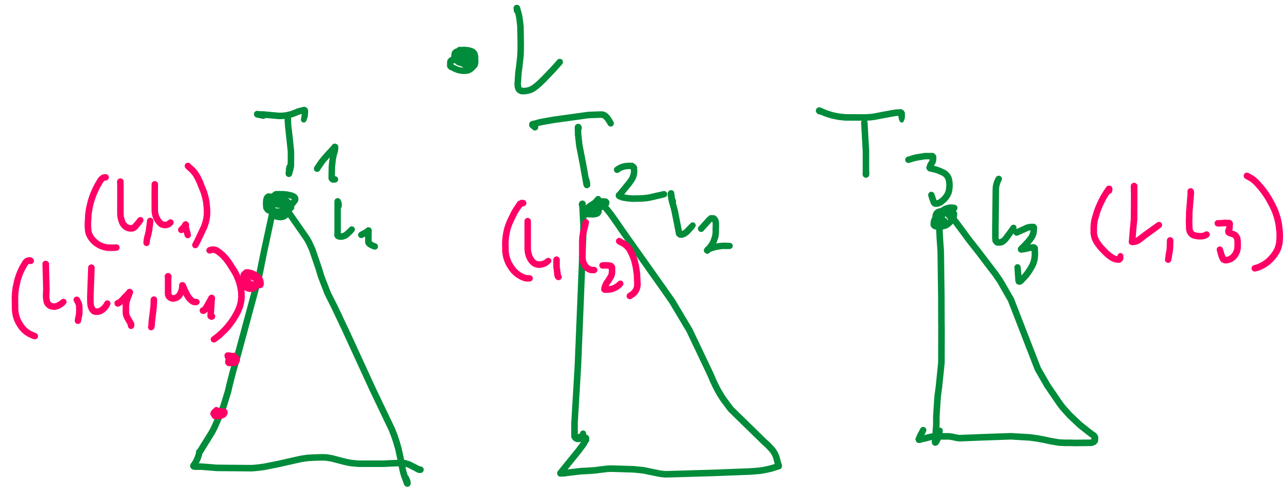
(2,10)

(5,?) if $2 \leq 5 \leq 10$

Naïve distance labeling

Algorithm 14.5 Naïve-Distance-Labeling(T)

Let l be the label of the root r of T
Let $T_1, \ldots, T_\delta$ be the sub-trees rooted at each of the δ children of r
for $i = 1, \ldots, \delta$ **do**
    The root of T_i gets the label obtained by appending i to l
    Naïve-Distance-Labeling(T_i)
end for



..

distance

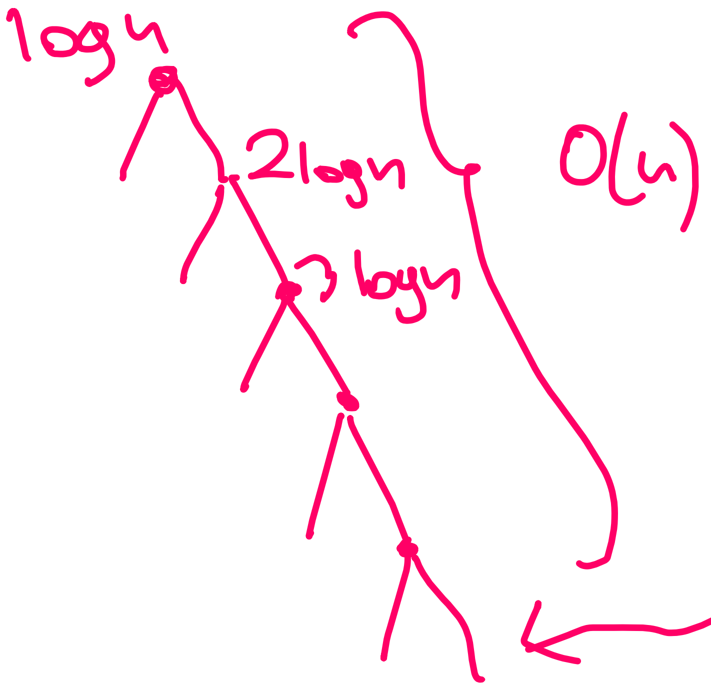
$(l, l_1, l_2, l_3, l_4, l_5$

$\;\|\quad\|\quad\|$

$(l, m_1, m_2, m_3, m_4, m_5, m_6)$

Naïve distance labeling

THM. This is an O(n log(n)) labeling



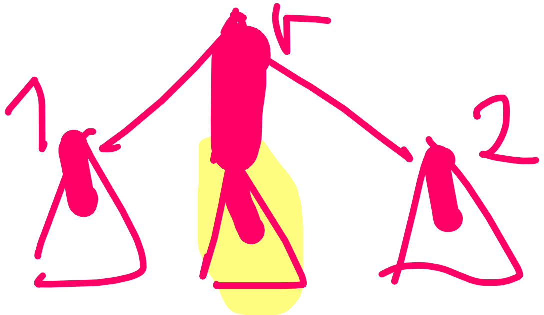
log n identifier

O(n log n)

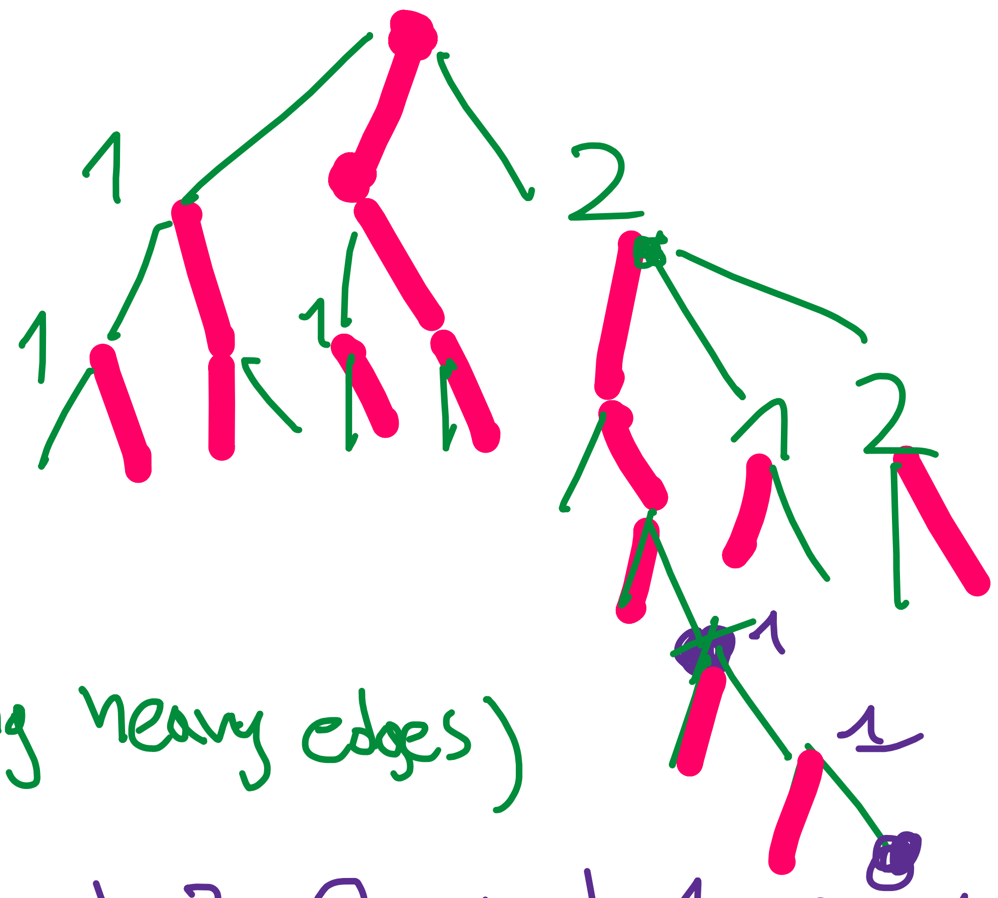
Heavy-light decomposition

Algorithm 14.7 Heavy-Light-Decomposition(T)

```
Node r is the root of T
Let T_1, ..., T_δ be the sub-trees rooted at each of the δ children of r
Let T_max be a largest tree in {T_1, ..., T_δ} in terms of number of nodes
Mark the edge (r, T_max) as heavy
Mark all edges to other children of r as light
Assign the names 1, ..., δ − 1 to the light edges of r
for i = 1, ..., δ do
    Heavy-Light-Decomposition(T_i)
end for
```



Heavy-light decomposition



(# edges along heavy edges)

(0, node 2, 2, node 1, 0, node 1, 0, node 1)

Heavy-light decomposition

THM. This is an $O(\log^2 n)$ distance labeling

1. heavy paths, connected via light edges

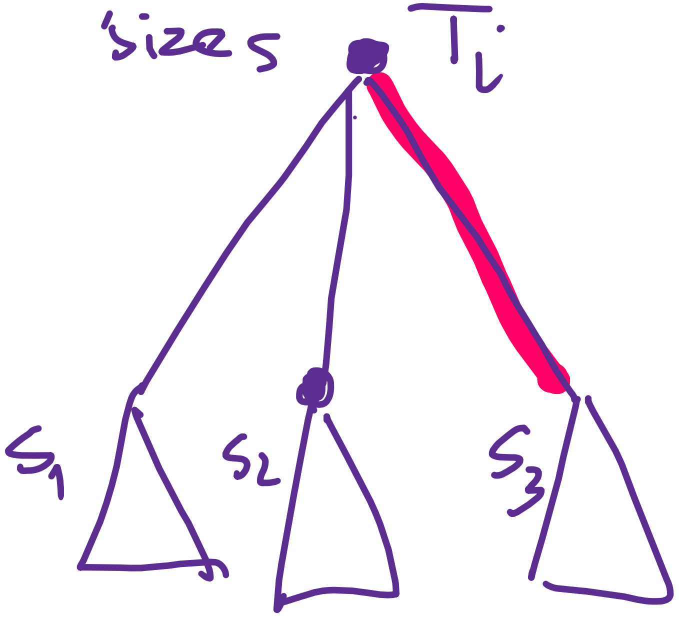
Heavy-light decomposition

THM. This is an $O(\log^2 n)$ distance labeling

1. heavy paths, connected via light edges

2. Lemma: The size of a subtree of a light node is at most ½ of the size of the parent

$s_2 \le \frac{s}{2}$



$s_1 \quad s_2 \le s_3$

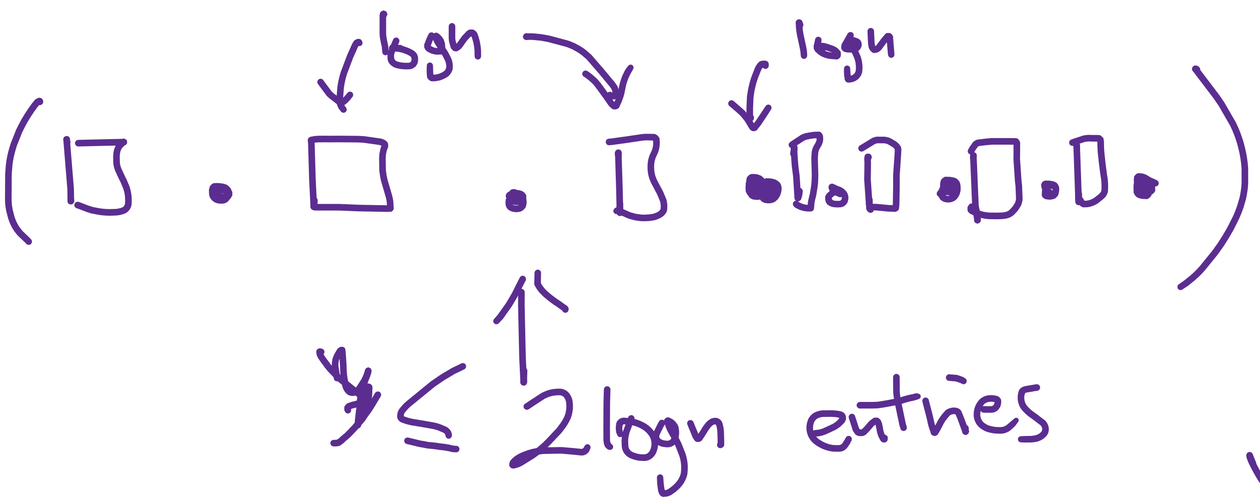
$s_1 + s_2 + s_3 = s$

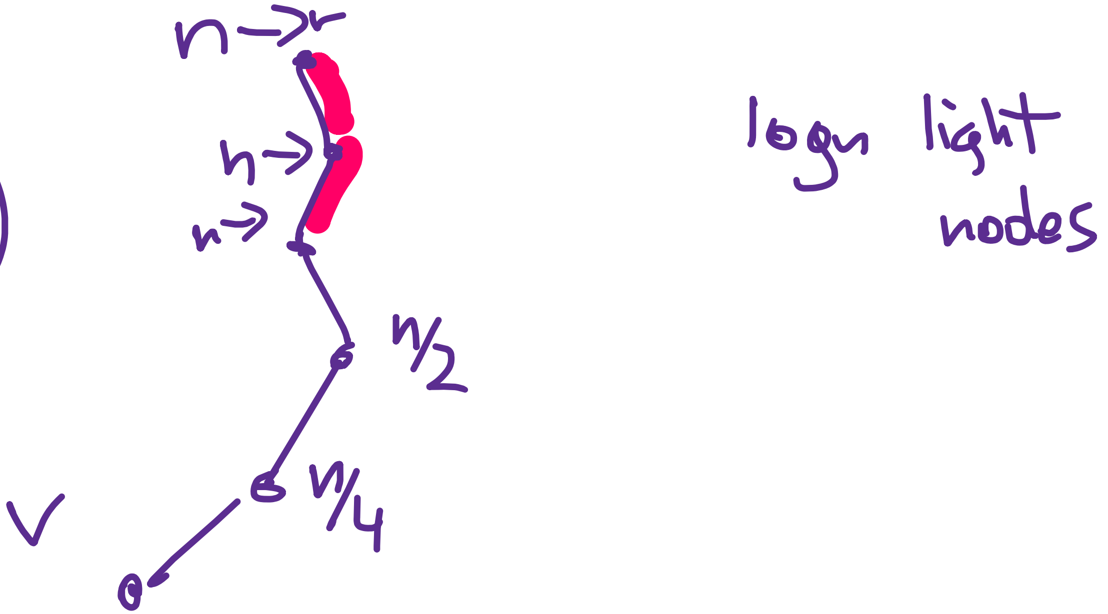
$\cancel{s_1} + s_2 \le s - s_3 \le s - s_2$

Heavy-light decomposition

THM. This is an $O(\log^2 n)$ distance labeling

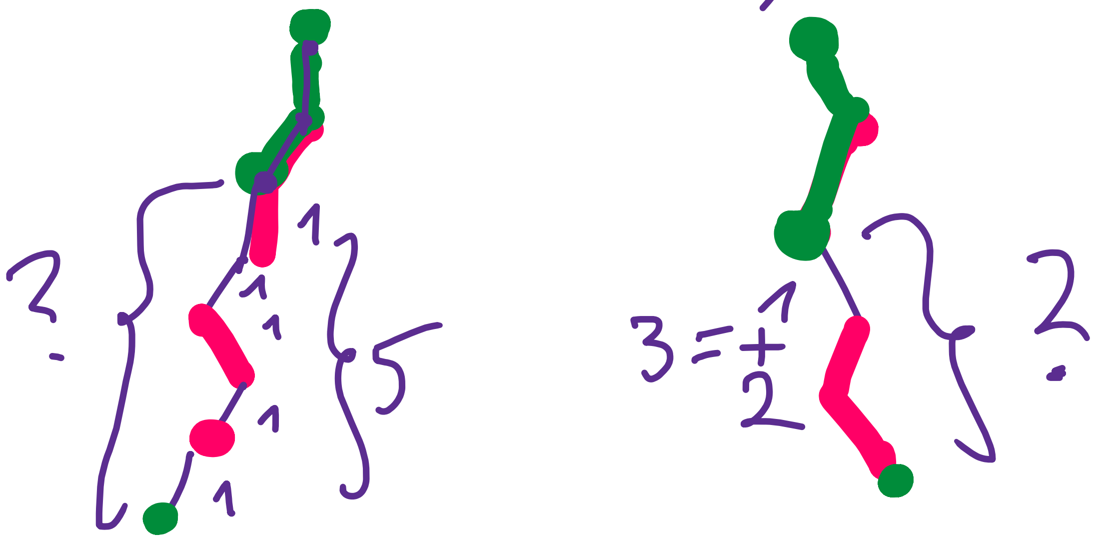
1. heavy paths, connected via light edges
2. Lemma: The size of a subtree of a light node is at most ½ of the size of the parent
3. Corollary: on a path at most log(n) light nodes





$$(\cdot\ \square\ \cdot\ \square\ \cdot\ \square\ \cdot\ \square\ \cdot\ \square\ \cdot\ \square)$$

$$(\cdot\ \square\ \cdot\ \square\ \cdot\ \square\ \cdot\ \square\ \cdot\ \square\ \cdot\ \square)$$



Road networks – navigation

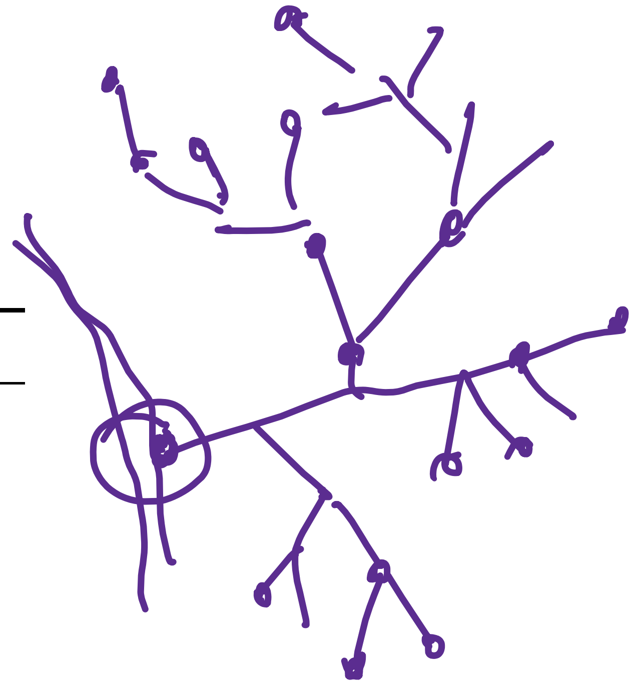
Task: For any two nodes compute their distance

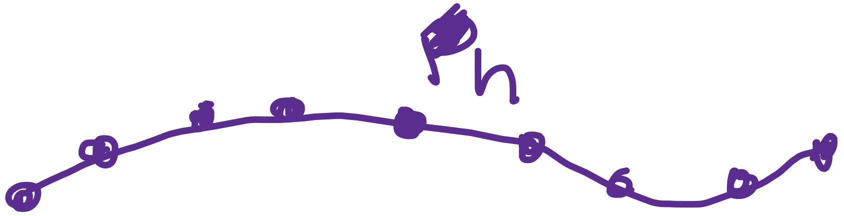
n – millions

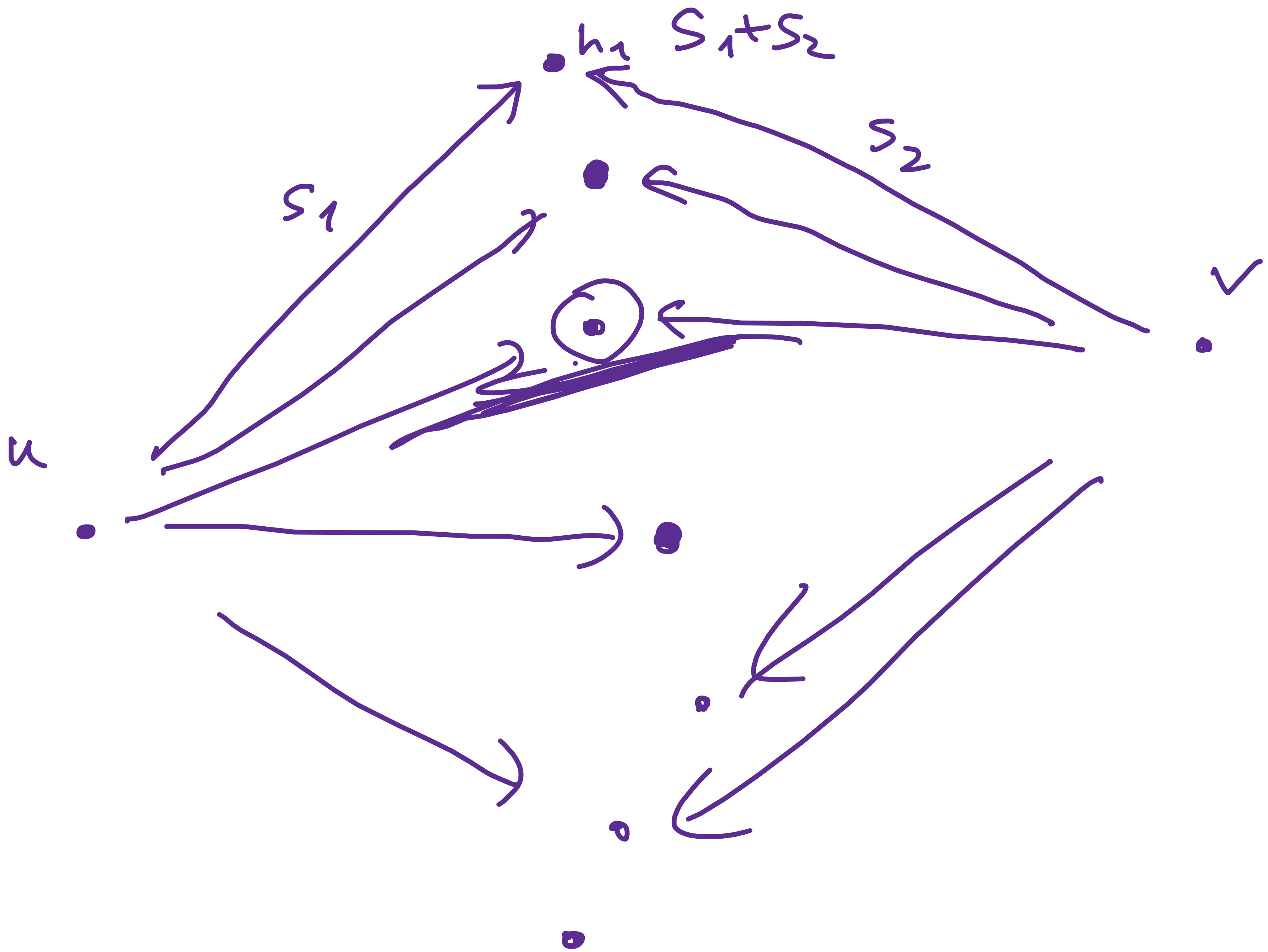
Road networks

Algorithm 14.9 Naïve-Hub-Labeling(G)

Let P be the set of all n^2 shortest paths
while $P \neq \emptyset$ **do**
  Let h be a node which is on a maximum number of paths in P
  for all paths $p = (u, \ldots, v) \in P$ **do**
    if h is on p **then**
      Add h with the distance $\mathrm{dist}(u, h)$ to the label of u
      Add h with the distance $\mathrm{dist}(h, v)$ to the label of v
      Remove p from P
    end if
  end for
end while







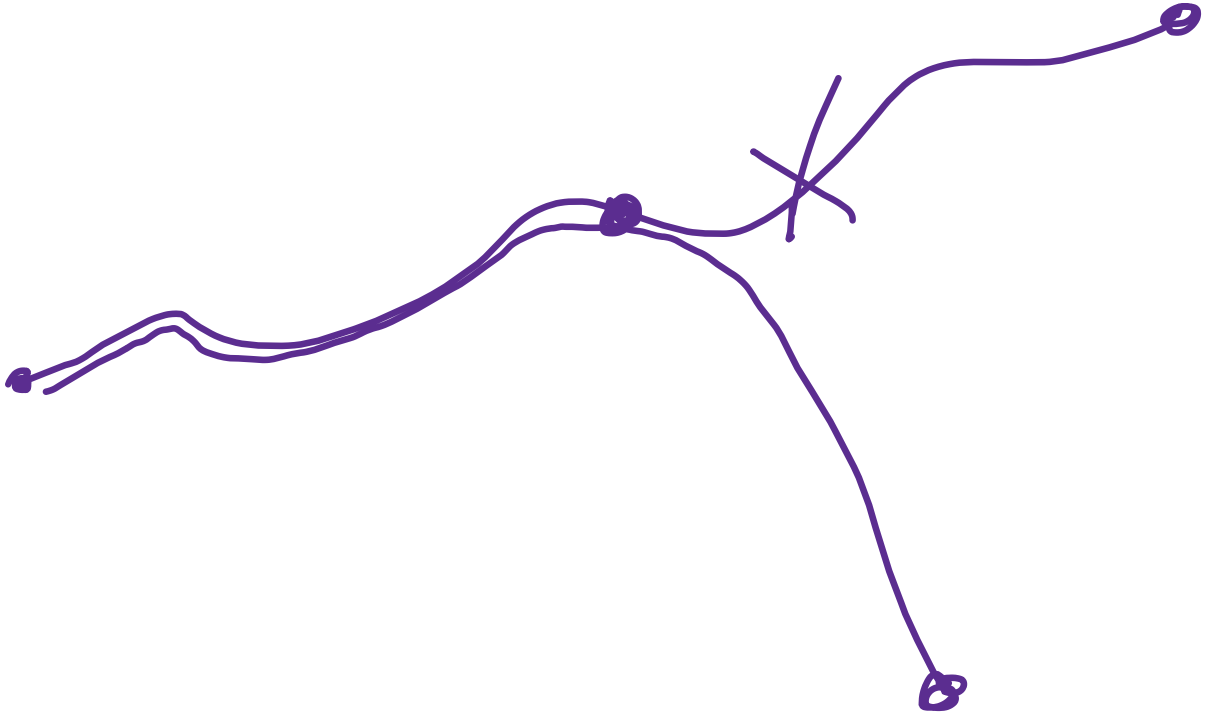
h_1
s_1+s_2
s_1
s_2
u
v

Road networks

Algorithm 14.9 Naïve-Hub-Labeling(G)

```
Let P be the set of all n^2 shortest paths
while P ≠ ∅ do
    Let h be a node which is on a maximum number of paths in P
    for all paths p = (u, ..., v) ∈ P do
        if h is on p then
            Add h with the distance dist(u, h) to the label of u
            Add h with the distance dist(h, v) to the label of v
            Remove p from P
        end if
    end for
end while
```

..



Shortest paths covers

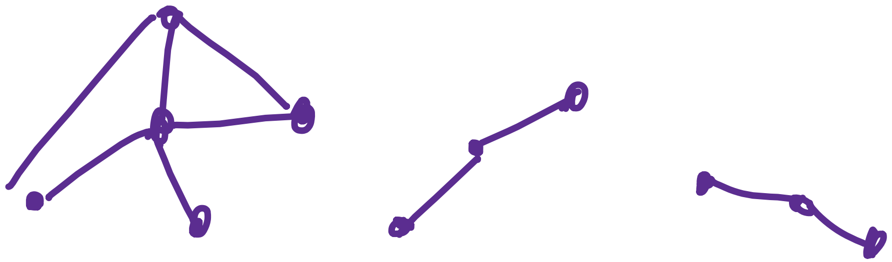
S is a shortest paths cover if it contains at least one node on each shortest path

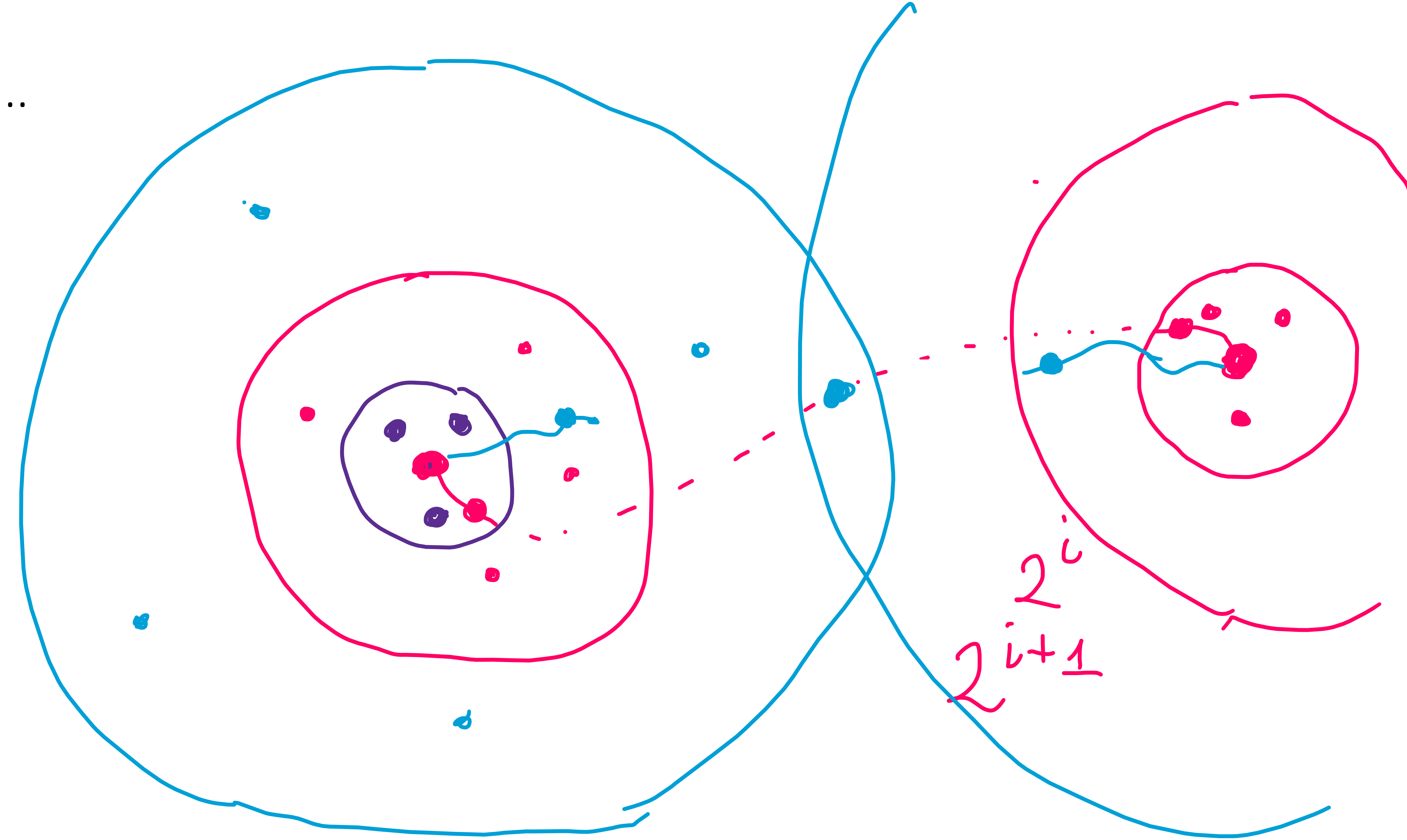
S_i is a cover that takes into account the paths of length between 2^{i-1} and 2^i

Road networks

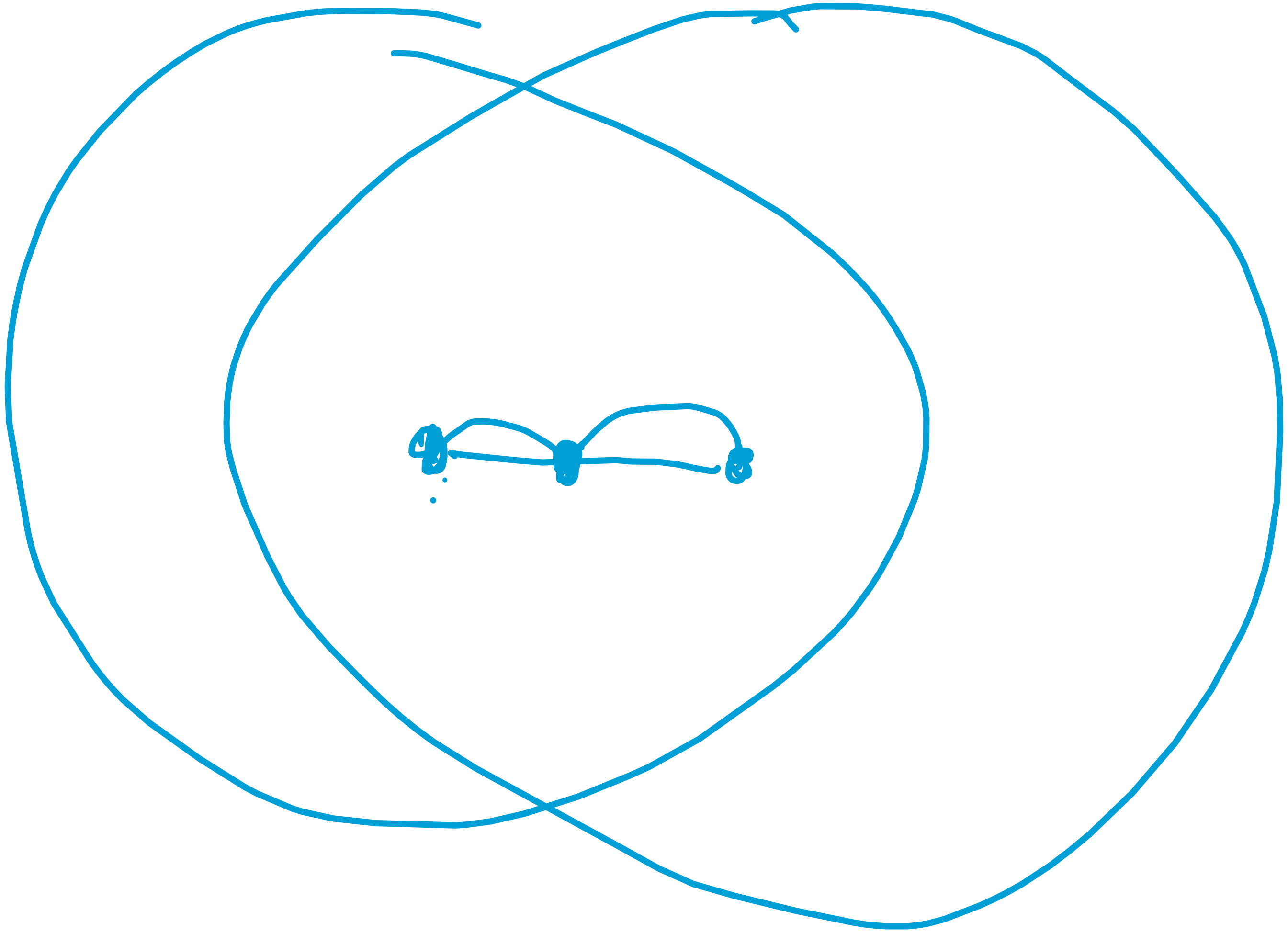
Algorithm 14.10 Hub-Labeling(G)

for $i = 1, \ldots, \log D$ **do**
    Compute the shortest path cover S_i
end for
for all $v \in V$ **do**
    Let $F_i(v)$ be the set $S_i \cap B(v, 2^i)$
    Let $F(v)$ be the set $F_1(v) \cup F_2(v) \cup \ldots$
    The label of v consists of the nodes in $F(v)$, with their distance to v
end for





2^i
2^{i+1}



..

Navigation opti—ization

road traffic ... — hard