① Weighted ε-nets

- runtime for fixed r:

$$6r\log\left(\frac{n}{r}\right)$$

- runtime for $r = 2^0, 2^1, 2^2, \ldots, 2r^*$

$$\sum_{i=0}^{\log_2(r^*)+1} 6\cdot 2^i \log\left(\frac{n}{2^i}\right) =$$

$\log\left(\frac{a}{b}\right) = \log a - \log b$

$= 2^{\log_2(r^*)+1}$

$\log_x(x^m) = m$

$$\sum_{i=0}^{\log r^*+1}\left(6\, 2^i \log n - 6\, 2^i \cdot i\right)$$

$$= 6\log n\left(2^{\log_2 r^*+2} - 1\right) - 6\left(2^{\log(r^*)+2}\log(r^*)+2\right)$$

$$= 6\log n\left(4r^* - 1\right) - 6\left(4r^*\log r^* + 2\right)$$

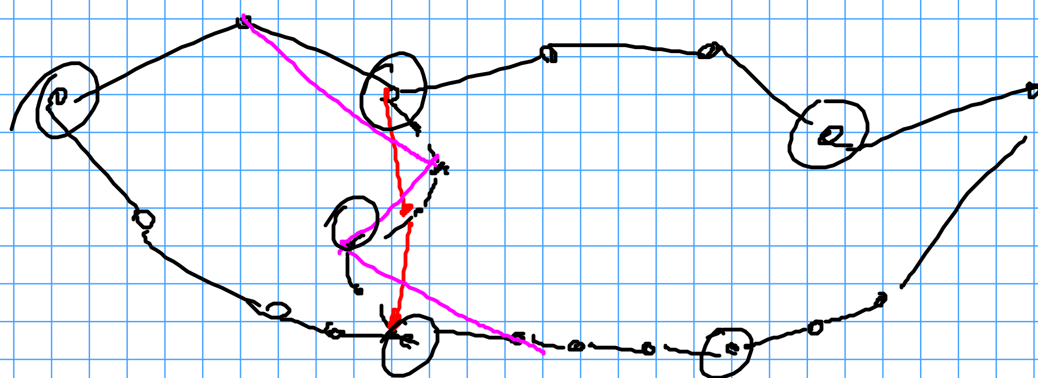
$$\leq 4\cdot 6r^*\log n - 4\,6r^*\log r^*$$

$$= 4\,6r^*\left(\log n - \log r^*\right)$$

$$= 4\,6r^*\log\frac{n}{r^*}$$

runtime for r^* known is

$6r^* \log \frac{n}{r^*}$ $\leadsto$ only a constant runtime overhead when guessing r^* iteratively

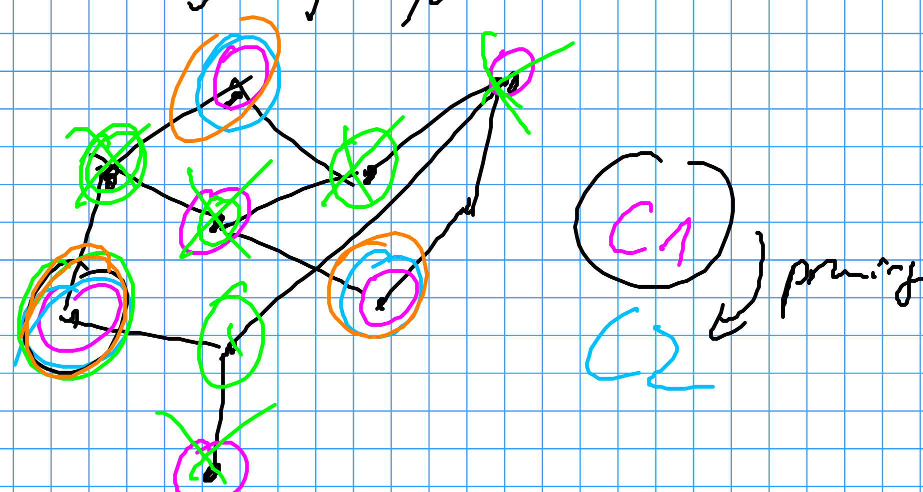
② 

$l_1, l_2, l_3, l_4 \ldots$

$C_1, C_2, C_3, C_4 \ldots$

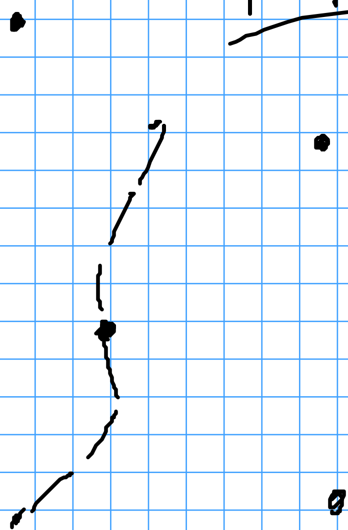
$C_{i+1} \subseteq C_i$

initial HS: alg. of your choice



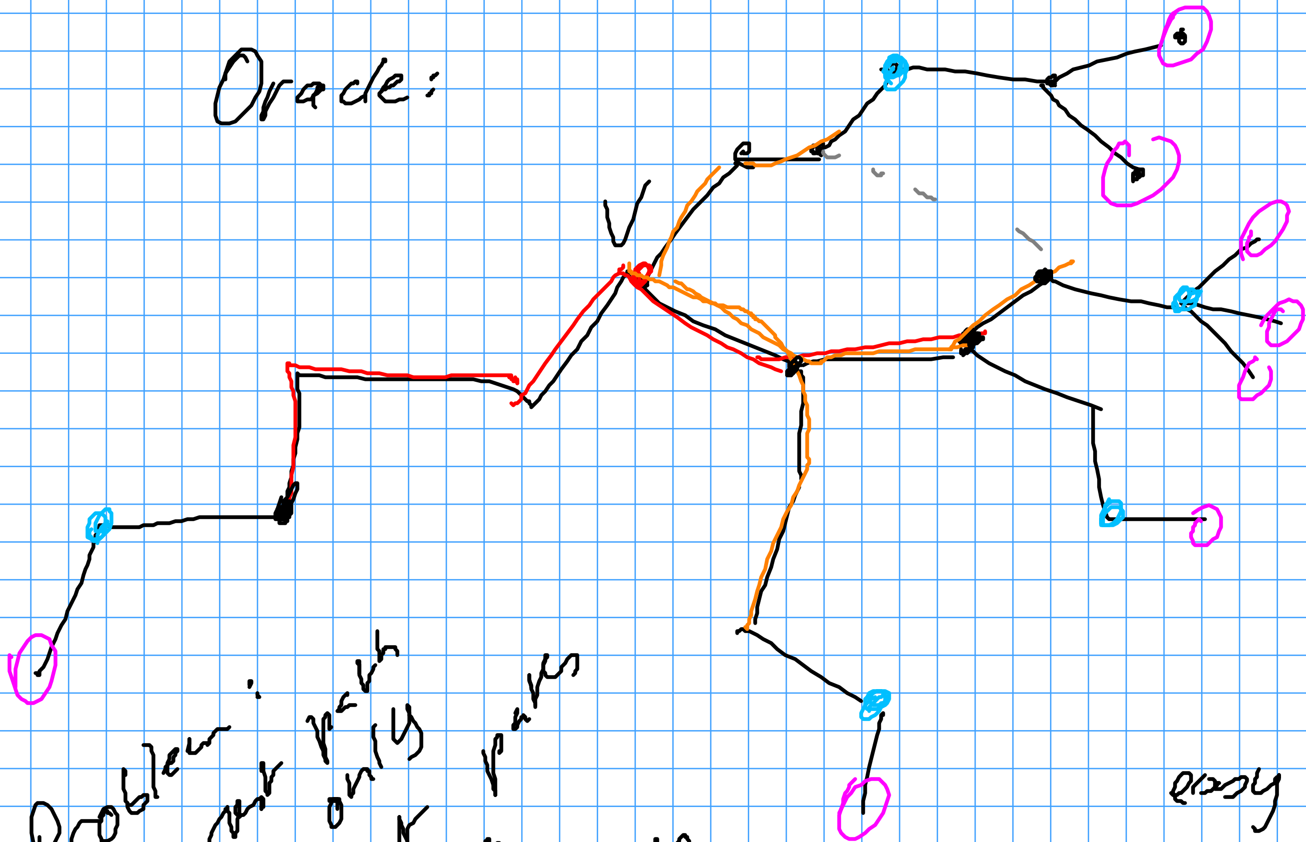
③ Pruning

Lemma: If we have an oracle that tells for $v \in C_{cur}$ if there exists a shortest path containing v as the only node from C_{cur}, then pruning return a <u>minimal</u> HS.



<u>Proof.:</u> It always maintains a feasible HS C_{cur}. Consider the moment v is regarded. If v is removed from the HS, nothing to do. If v is maintained, i.e. in the graph exists a shortest path only hit by v and no other node in the final HS. If we would remove v from the final HS, then it would not be a HS any more.

Oracle:



k

$k/2$

Problem: Shortest path tree only tells about paths with V being the source

2) sho. paths via V might be missed

easy fix:
Shortest path tree with $k/2$
& if any comb. of paths of length k is unit, maintain v

BUT: false pos. possible

One possibility:

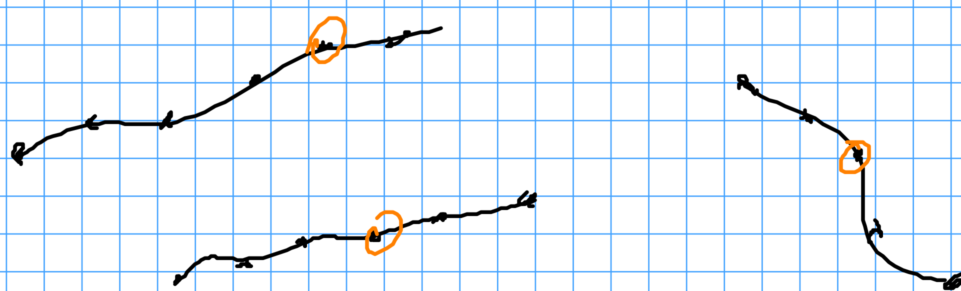
- APSP
  problem:

Another poss.:

- run a Dijkstra from all nodes in the shortest path tree until all other nodes in the tree are settled

Lower Bounds for HS

- best node hits q paths and we have Q paths in total
  ↳ $\left\lceil \frac{Q}{q} \right\rceil$
- maximal set of independent paths



- on-the-fly LB (without set extraction)
  - path set $P = \emptyset$
  - run Dijkstra from random source s
  - if in the shortest path tree there is a path p of required length not intersecting any path in P, then add p to P
- $LB = |P|$