1 MINIMUM SPANNING TREE

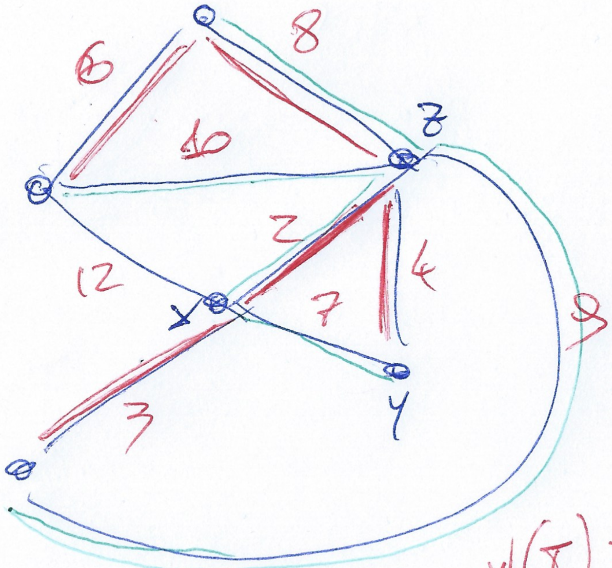
WEIGHTED GRAPH CONNECTED

G UNDIRECTED

ACYCLIC CONNECTED SUBGRAPH

$n-1$ ARCS

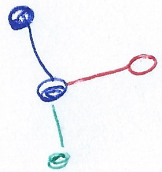
BY INDUCTION



$W(T) = 23$

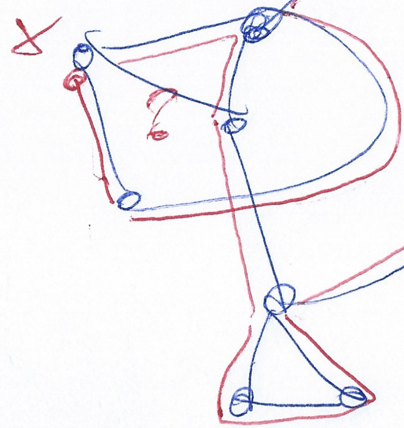
$W(T) = 36$

NOT MINIMUM



EULERIAN TOUR

VISIT ALL EDGES ONE TIME



- LOOK AT ALL POSSIBLE SPANNING TREES ?

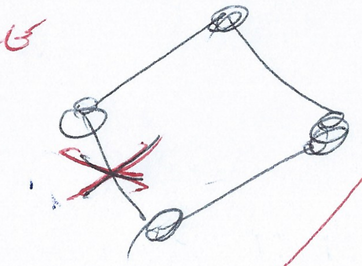
n^{n-2} DIFFERENT SPANNING TREES

2 CUT RULE

CYCLE RULE DELETION RULE

↓

DELETE THE MAXIMUM EDGE IN A CYCLE

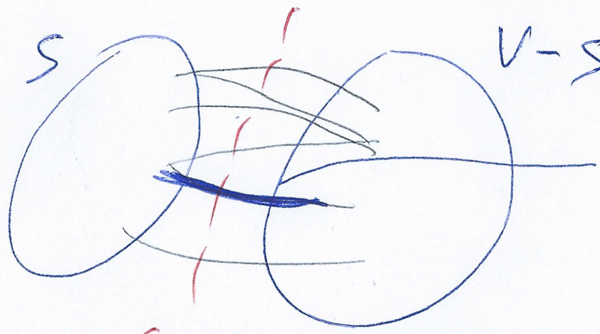


ADDITION RULE

CUT $G(V, E)$

$S \subseteq V$

S V−S



TAKE THE MINIMUM

CUT

AT LEAST ONE EDGE FROM THE CUT IS IN M.S.T.

$E \cap S \times (V - S)$

GREEDY ALGO FOR M.S.T

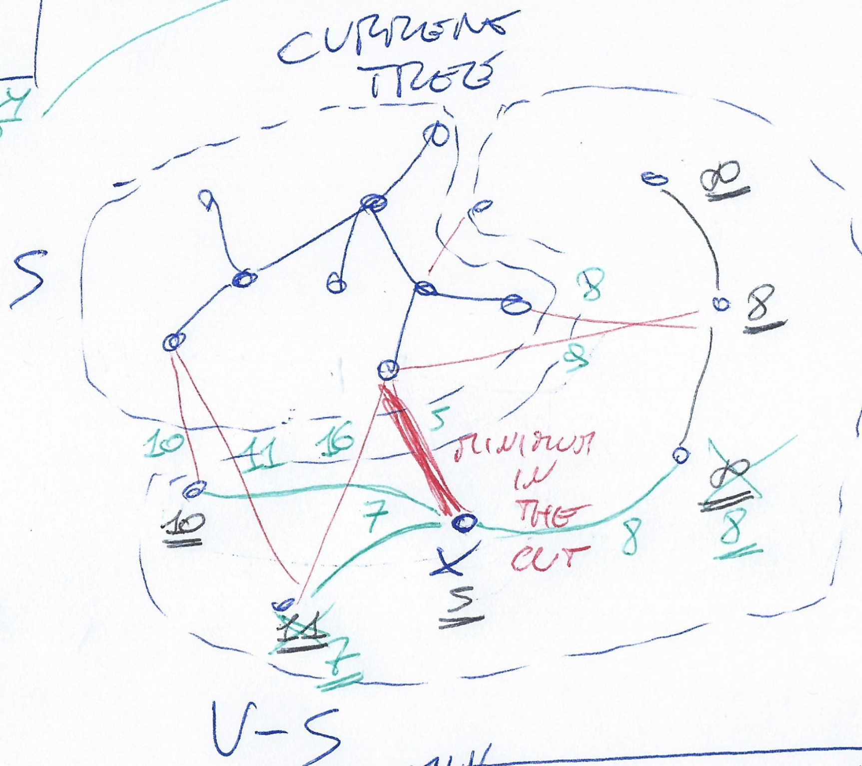
$T = \emptyset$

WHILE T IS NOT CONNECTED

- CHOOSE A CUT DISJOINT FROM T (NO EDGES FROM T IN THE CUT)
- TAKE THE MINIMUM EDGE IN THE CUT $\longrightarrow T$

③ PRIM'S ALGORITHM FOR M.S.T.

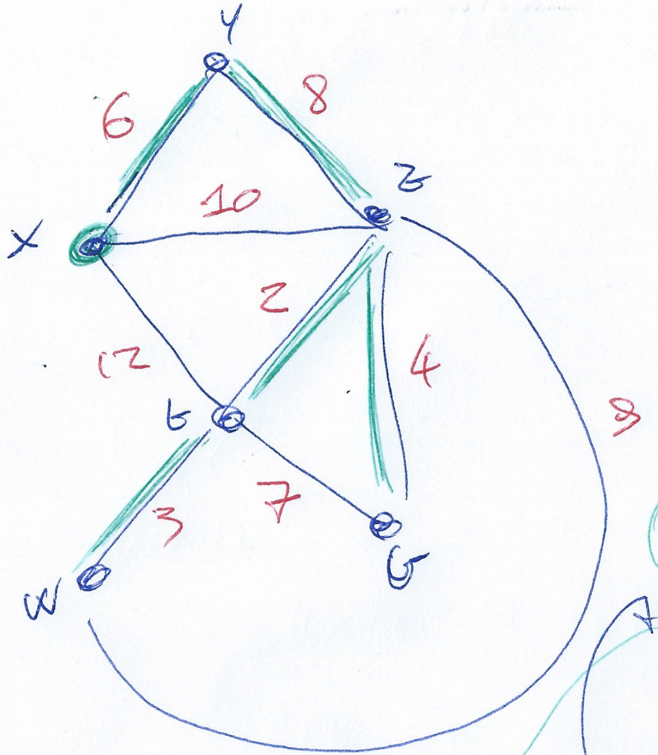
→ $O(m \cdot \lg n)$ BINARY HEAP



FIBONACCI HEAPS

$O(m + n \lg n)$

UP TO n^2



$O(n)$

- BUILD A PRIORITY QUEUE Q with all vertices ∞ PRIORITY, NO PARENT
- TAKE VERTEX ⓧ FROM Q
- FOR EACH e incident on x $e(x,y)$

  if y IS NOT CONNECTED TO THE TREE

  if $w(x,y) < p(y)$

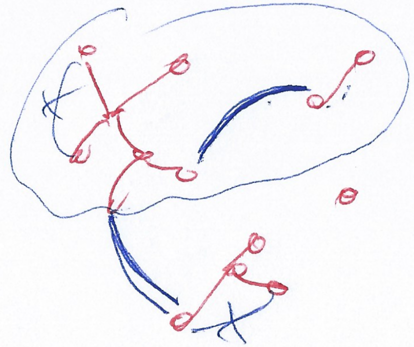
  SET PRIORITY(y) TO $w(x,y)$

  $O(\lg n) \to$ **m** TIMES

$O(\lg n)$ n TIMES

4 KRUSKAL'S ALGORITHM M.S.T.

→ AS PRIM'S



- SORT EDGES
  INCREASING WEIGHT
- T = ∅ ← FOREST (SET OF TREES)
- FOR EACH EDGE e
  - IF e FORMS A CYCLE
    (JOINS TWO VERTICES IN THE SAME COMPONENT OF T)

    DISCARD

    ELSE

    ADD e TO T

    JOIN THE TWO C.C.

5 SHORTEST PATHS

WEIGHTED GRAPH $G=(V,E)$
(DIRECTED)

- FROM x TO y — SINGLE PAIR

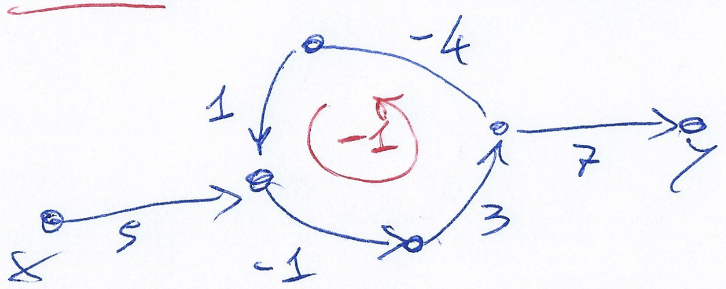
SAME COST

- FROM x TO y FOR ALL y — SINGLE SOURCE SHORTEST PATHS | S.S.S.P.

$W(e) \geq 0$
POSITIVE WEIGHTS
$\Downarrow$
NO NEGATIVE CYCLES

- FROM EACH x TO EACH y — ALL PAIRS

BY n TIMES SSSP
↓
BETTER ALGOS



6 DIJKSTRA's ALGORITHM

S.S.S.P.

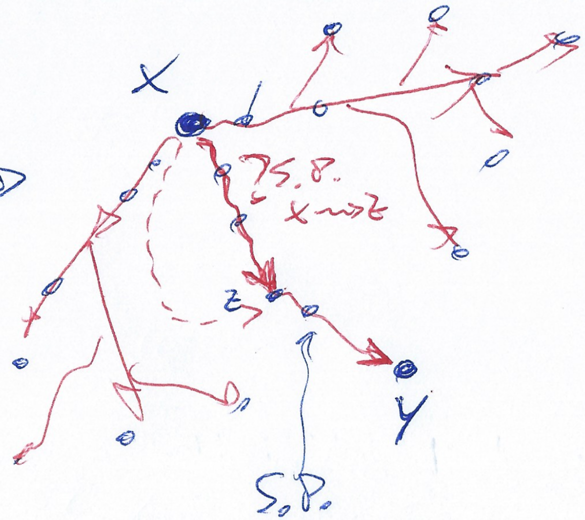
(POSITIVE WEIGHTS) DIRECTED

↓

S.S.S.P. TREE

ADD A LEAF AT A TIME

$O(m \cdot \log n)$

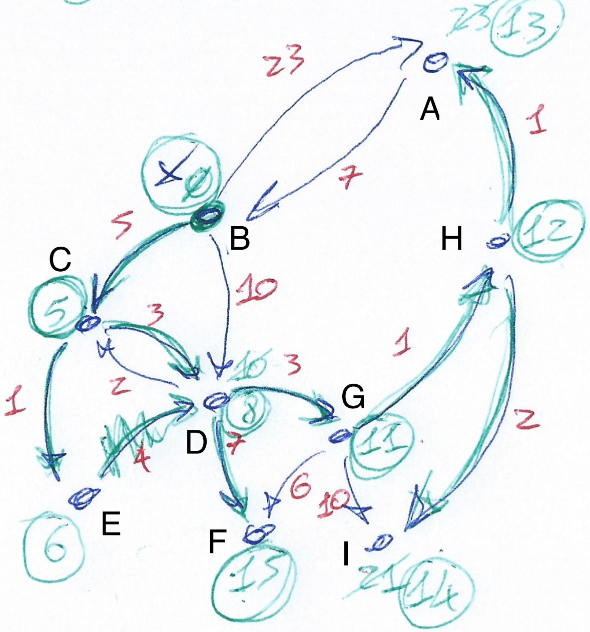


X ~> Y

↓

ALL PREFIXES OF X ~> Y ARE SHORTEST PATHS

O ~~REAL TRUE~~ DISTANCE



EACH VERTEX HAS DISTANCE ∞ FROM X

→ REACH VERTEX Y DISTANCE d