

- 1) L has at the string that contains the language
- 2) validate if L contains some item ~~that~~ equals to S, after to convert to binary

Algorithm reduce (L)

for each item to L do

if $\text{convert}(\text{item}) == S$

return ~~false~~ true

return false.

A. ① Randomly we pick a spanning tree of G and push it in a sequence T

② We need to write an algorithm for calculate the sum of the subset, is Exist a MST with at least min Weight.

Algorithm $MSTtoSS(G, T, min)$

sum = 0

foreach e to $T.elements()$ do

sum = sum + getWeight(e)

If sum \geq min

return Yes

else

return NO_SOLUTION

③ Randomly we are going to put the ~~new~~ shortest path in T

② We need to write an algorithm for calculate the MST of T with weight at ~~most~~ most max does exist?

Algorithm SP to MST (G, T, \max)

if $T.size() \geq G \text{ num vertices} - 1$ then
return NOT_FOUND

total $\leftarrow 0$

for each e in $T.elements()$ do

set $\mathcal{E} \text{ of } T(e, \text{yes})$

total $\mathrel{:=}$ total + weight(e)

(isConnected, isCycle) \leftarrow BFS(G)

if isConnected \wedge ! isCycle \wedge total $\leq \max$ then

return yes

else NOT_FOUND

Algorithm initResult(G)

components $\mathrel{:=} 0$ isCycle $\mathrel{:=}$ false

Algorithm preComponentVisit(G, v)

components \leftarrow components + 1

Algorithm preEdgeVisit(G, v, e)

if setEdgeOfT(e) \neq yes then

setLabel(e, CRP)

Algorithm crossEdgeVisit(G, v, e, w)

isCycle $\mathrel{:=}$ true

Algorithm result(G) return (components, isCycle)