14/03/2024

CONTINUO SU M.S.T: ALGO. DI PRIM

PROBLEMA: SEMPRE IL M.S.T. (MINIMUM SPREADING TREE).
(SEE SLIDE PRIMA).

prima abbiamo visto l'algoritmo di kruska, ora vediamo quello di prim.

> N.B. ANCHE QUI LA CUT PROPERTY E LA CYCLE PROPERTY SONO IMPORTANTI x DIM. CORRETTEZZA.

ALGORITMO

- **IDEA**: PRENDO UN QUALSIASI NODO r (IN GRAFO G) = RADICE of TREE T. T CRESCE GRADUALMENTE, ADD e = EDGE $\in$ CUTSET + PICCOLO
- **CORRETTEZZA** → x $n-1$ VOLTE, VIENE SFRUTTATA LA CUT PROPERTY.
- **PSEUDOCODICE**

  1° IDEA: per n volte, vedo tutti gli archi e prendo il minimo di quelli che attraversano il taglio.

  COSTO = $O(m \cdot n)$ → VEDO ALMENO TUTTI GLI ARCHI (m) AD OGNI NODO (n)
  → FUNZIONA MA È MOLTO INEFFICIENTE

  2° IDEA + EFF. → USO S.D. CODA PRIORITÀ:
  - SET S = NODI RAGGIUNTI
  - CODA Q = NODI NON ANCORA RAGGIUNTI
    → PRIORITÀ = COSTO ARCO.
  - $a[v]$ = COSTO CHEAPEST ARCH TRA v e S
    (v → NODO)

PSEUDOCODICE

2° IDEA

TUTTI COSTO MAX TRANNE 2

ALL NODE INSERT INCLUSO $S \to a[S] = 0$,

```
Prim(G, s) {
   foreach (v ∈ V) a[v] ← ∞
   a[s] ← 0   --> RADICE, QUINDI a[ROOT] = 0
   Initialize an empty priority queue Q
   foreach (v ∈ V) insert v onto Q with priority a[v]
   Initialize set of explored nodes S ← φ  --> S
   Initialize T to the tree containing only s.  --> T

   while (Q is not empty) {
      u ← delete min element from Q  ] NODO u COSTO + PICCOLO ESPLORATO
      S ← S ∪ { u }
      foreach (edge e = (u, v) incident to u)
         if ((v ∉ S) and (c_e < a[v]))
            make u parent of v in T
            decrease priority a[v] to c_e
   return T
}
```

COSTO

- $O(n+m)$ = OP. SENZA CONS. OP. COD. PRIORITÀ
- C. PRIORITÀ = n INS., n MIN. DEL., m DECREASE KEY
  - ↳ DIPENDE CHI USO:
    FIBONACCI = $O(m + n \lg)$

$O(m + n \lg n)$

APPLICAZIONE MST: CLUSTERING

cosa è il clustering?

PROBLEMA CLUSTERING:

DEF. INFORMALE - dati degli oggetti, si vuole raggupparli in 'cluster', dove gli oggetti in cluster = sono simili tra loro.

DEF FORMALE: GRUPPO OGG. = CLUSTER / GRUPPO DI GRUPPI = CLUSTERING.

- CLUSTERING : insime U di n oggetti, classificati in gruppi coerenti.

- FUNZ. DISTANZA : valore numerico di 'vicinanza' tra due oggetti

- PROBLEMA : raggruppare gli oggetti in cluster, con ogni ogg in un cluster hanno val funzione entro una certa soglia.

- K-CLUSTERING : / OGG. IN K CLUSTER

- FUNZ. DEF FORMALE : $d(P_i, P_j) = 0$ IF $P_i = P_j$ (UGUALI)

  $d(P_i, P_j) \geq 0$ (NON NEGATIVO, LONTANI)

  $d(P_i, P_j) = d(P_i, P_j)$ SIMMETRICA

- SPACING → MIN. D. TRA 2 PUNTI OF 2 CLUSTER

- OBB → MADE K-CLUSTER U MAX SPACING.

ALGO/

IDEA → GREEDY ALGO. ∧ PROPIETA' MST (VEDREMO)

PROCEDIMENTO

1. CREO GRAFO CON SET U ⟹ SET DI NODI
2. 2 CLUSTER + VICINI E CI AGGIUNGI UN 'ARCO
3. REPEAT PER n-K VOLTE, COSI HO PRECISAMENTE K CLUSTER.

CORRETTEZZA

PRE-OSSERVAZIONE : • L'ALGO È EFFETTIVAMENTE L'ALGO DI KRUSKAL

LA ≠ È CHE SI FERMA DOPO n-K VOLTE

• CLUSTER DI k = 1 ⟹ MST CON HIERCAL CLUSTER

↳ HIERCAL CLUSTER
=
MST U CLUSTER x OGNI VALORE $k = 1, 2, \dots, n-1, n$

- UN k-CLUSTER POSSIAMO VEDERLO COME UN MST A CUI VENGONO TOLTI I k-1 ARCHI PIÙ COSTOSI

DA QUESTO

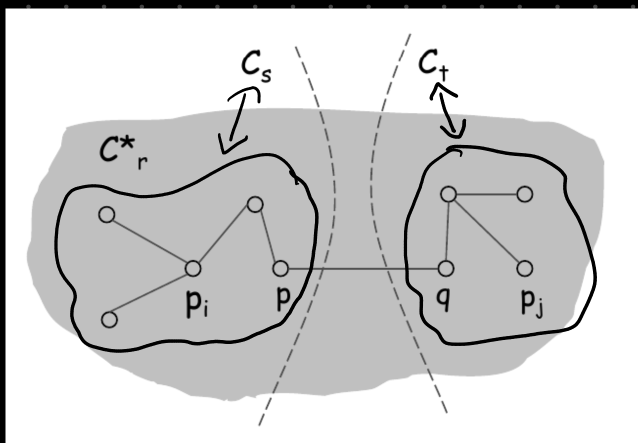
TEOREMA:

SIA C^*, UN k-CLUSTER DOVE $C_1^*, C_2^*, \dots, C_k^*$ SONO FORMATI ELIM. I k-1 ARCHI + COSTOSI NEL MST.

C^* È UN k-CLUSTER DI MAX SPACING

DIM → CAMBIO ARGOMENTO

- CONSIDERIAMO ANCHE UN C CHE CONTIENE ALTRI CLUSTER $C_1, C_2, C_3, \dots, C_k$



- d^* È SPACING = k-1 ARCO + COSTOSO DI MST
- SIA p_i, p_j (NODI) BE IN = CLUSTER DI C^* (C_r^*), MA IN ≠ CLUSTER DI C (C_s, C_t)
- IN C^* CI SARÀ UN ARCO (p,q) CHE 'DIVIDE' C_s e C_t NEL PATH p_i, p_j
- ∀ ARCO IN PATH $p_i - p_j$ HA LEN $\leq d^*$ (DATO CHE KRUSKAL LI HA SCELTI, INVECE p-q ?
- p-q HA COMUNQUE LEN (QUINDI SPACING) $\leq d^*$ → NON SONO I CLUSTER MIGLIORI

→ DATO CHE SONO IN 2 CLUSTER

↳ QUINDI