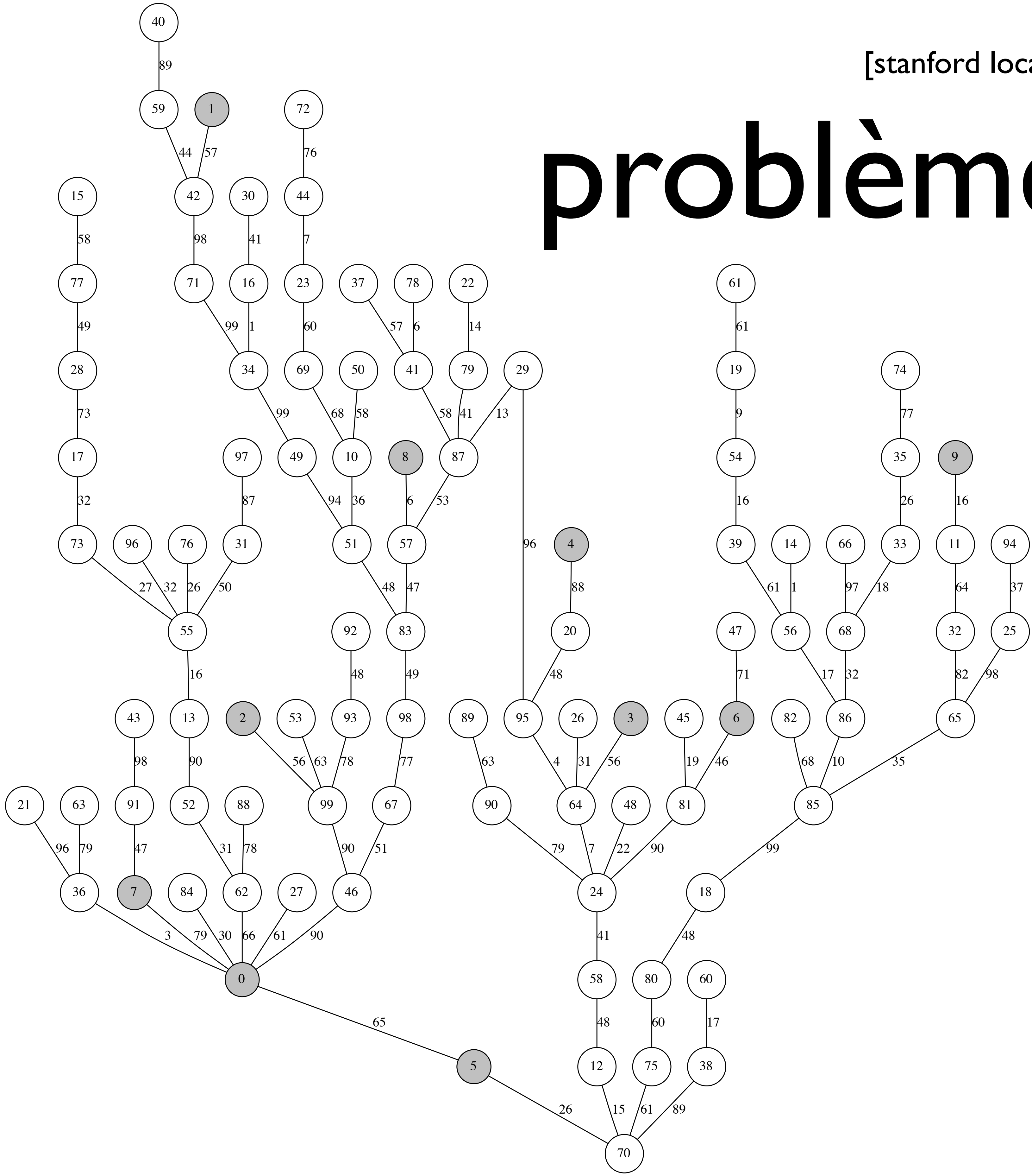
Petit problème algorithmique

séminaire S — 25 Sep 2015

[stanford local programming contest 2011 *city driving*]

problème



Entrée

un graphe connecté pondéré à n sommets et n arêtes

une liste de $\sqrt{n}$ sommets

Sortie

distance entre tout couple dans cette liste.

Complexité attendue

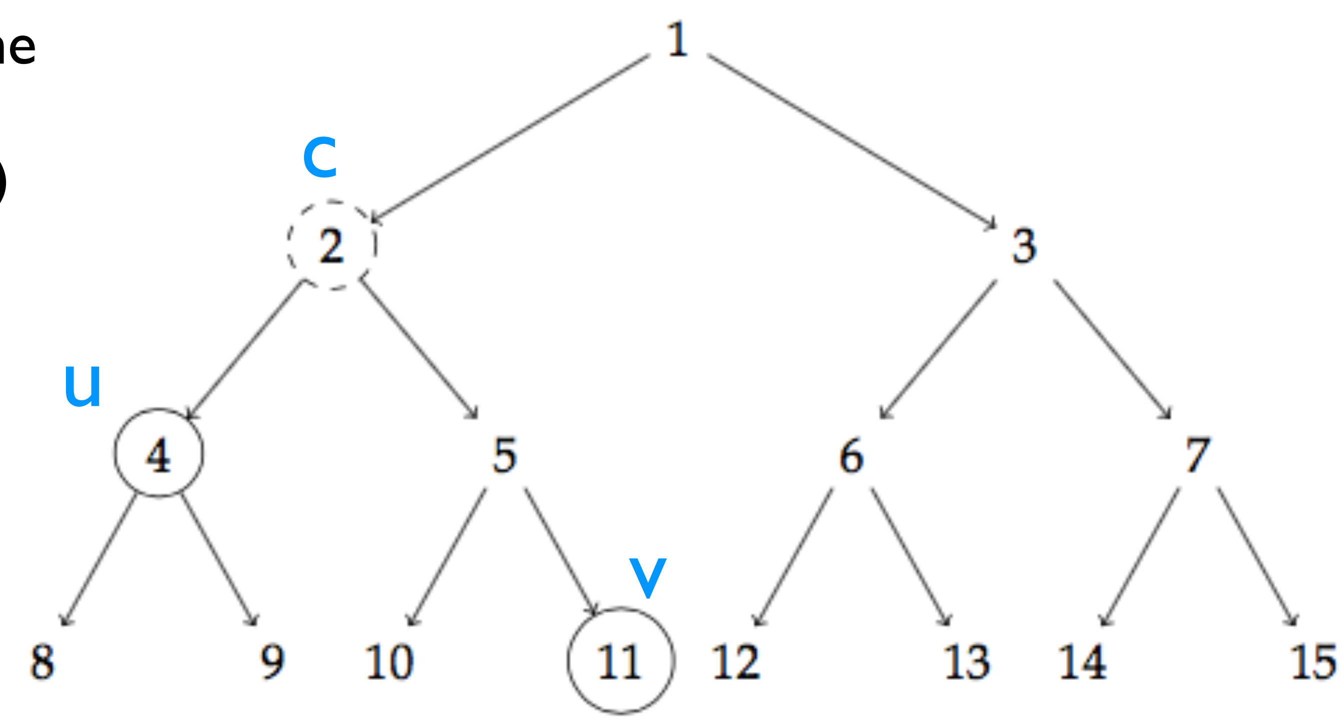
O(n log n)

arbre + arête (a,b)

- si on enlève une arête (a,b) de l'unique cycle on a un arbre
- Si d est la distance dans l'arbre, alors la distance dans le graphe entre u et v est
  min{ d(u,v),
  d(u,a)+w(a,b)+d(b,v),
  d(u,b)+w(b,a)+d(a,v) }
- on accepte un facteur 3 dans la complexité et on se restreint au calcul de distances dans un arbre

enraciner l’arbre

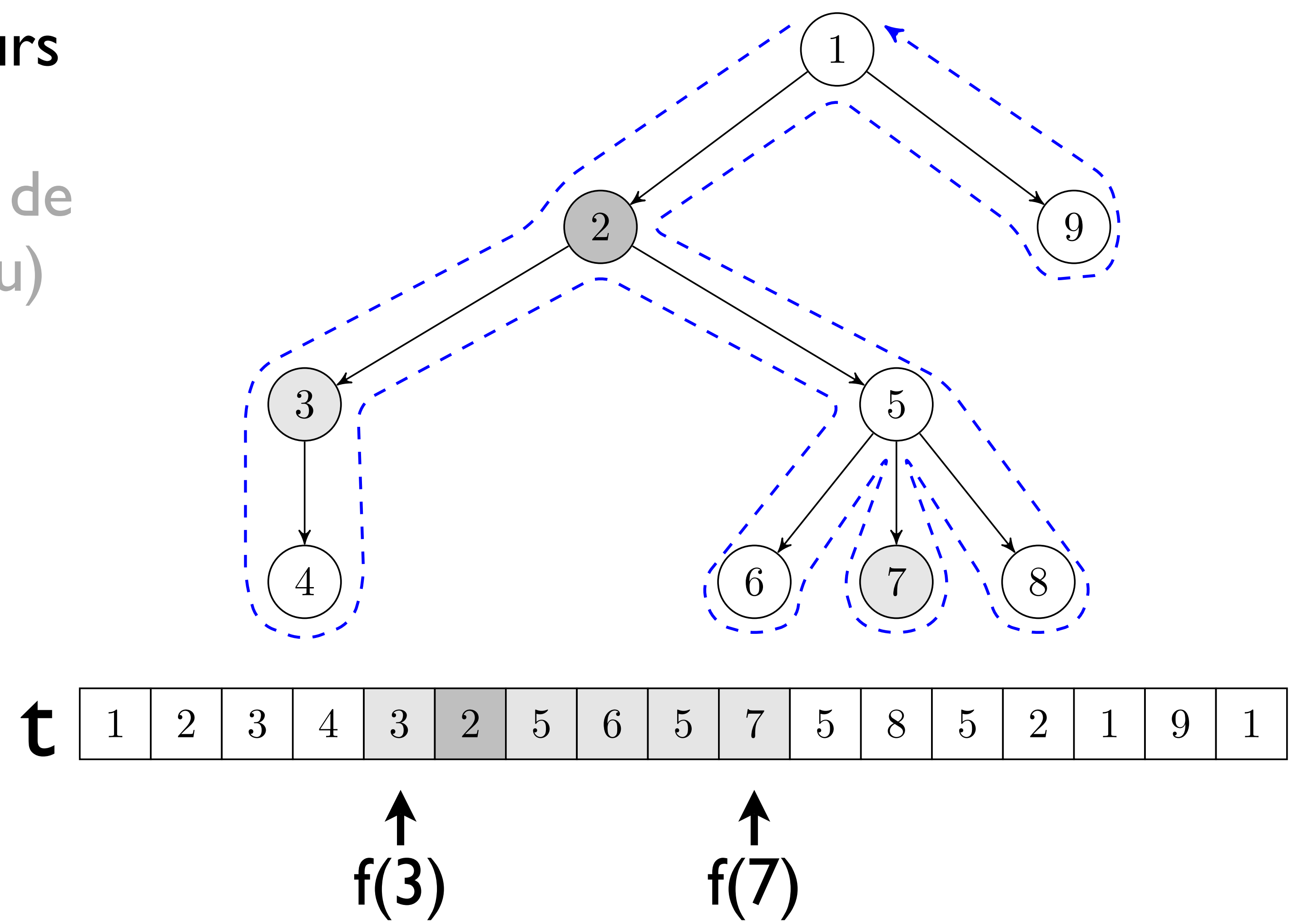
- on choisit un sommet abitraire comme racine
- ceci oriente l’arbre (antécédent, descendant)
- Pour tout sommet v, soit g(v) := d(racine,v).
- On peut calculer g par programmation dyn.
- Pour u,v il existe un antécédent commun le plus proche, disons c.
- Alors d(u,v) = d(u,c) + d(c,v)
  = g(u)+g(v)-2*g(c)
- comment calculer rapidement l’antécédent commun le plus proche ?




```
entrée: 4, 11
sortie: 2
```

réduction au problème du minimum dans une plage

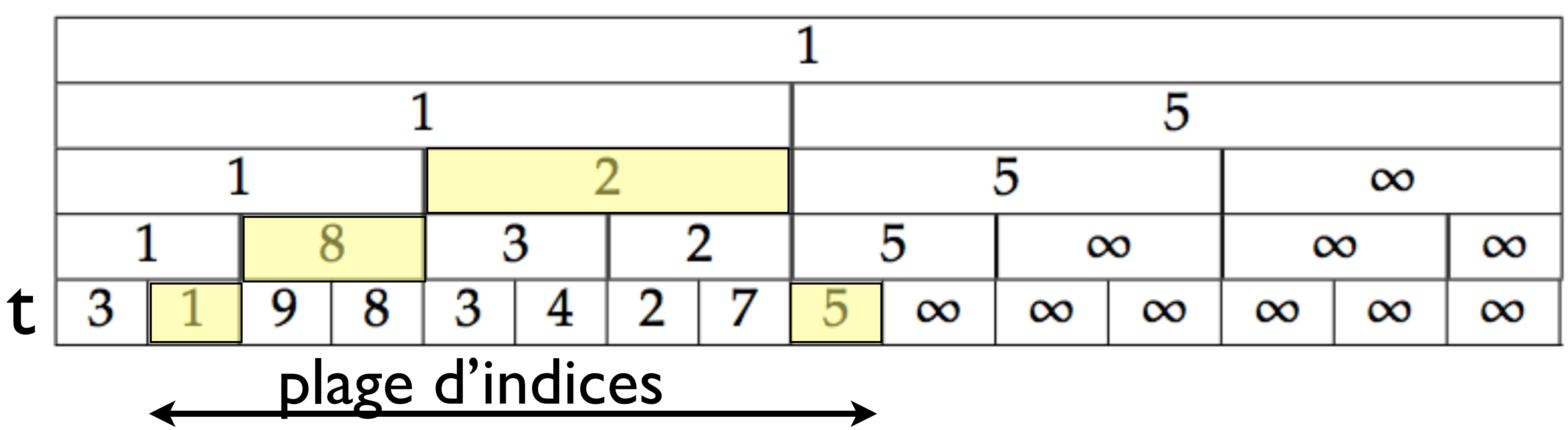
- écrire dans t la trace d'un parcours en profondeur
  (écrire u en début de traitement de (u,v) et à la fin du traitement de u)
- noter f(v) la date de fin de traitement de v
- le sommet c entre f(u) et f(v) de plus petite hauteur est l'ancêtre commun le plus proche



Simplification de la figure: on a supposé que père[v] < v, alors on peut omettre la hauteur du sommet v

minimum dans une plage

- Construire un arbre binaire complet avec les élements de t aux feuilles (compléter t avec ∞, pour que long. soit puiss2).
- Écrire dans chaque noeud le minimum de ses deux fils.
- Construction: O(n)
- Requête: O(log n)



résumé

- problème initial
- *3 réduction au problème de distances dans un arbre
- +n réduction au problème d'ancêtre commun le plus proche
- +nlogn réduction au minimum dans une plage
- complexité O(n log n)