partie de GREF

Bashar DUDIN

Disclaimer

Ce sont des notes de cours imparfaites qui n'ont absolument rien d'officiel. Le professeur n'a eu strictement aucune implication dans la création de ce fichier. Il n'engage personne et en particulier ni le professeur ni les étudiants qui l'ont écrit.

C'est juste des élèves quelconques qui font de leur mieux pour suivre en cours et vous partager leurs notes pour vous aider dans vos révisions.

Sur ce, bon travail :).

Table des matières

1 Infos

You can get many infos there. https://github.com/bashardudin/GraphsAndFlows

2 Recap

shortest path

- single-pair (svg dst) shortest path problem
- single src any destination : shortest path problem
- Any pair shortcut path pb

algos

- Bellman-Ford [general no negative valued edges]
- Dijkstra [positive weights]
- Bellman [no cycles]
- Floyd warshall

2.1 Safest path

Aim : safest path along which to go.

To use shortest paths algorithms for it is about apllying shortest paths to graphs having -log(above weights).

2.2 Finding paths

To find critical paths going back from ending point you look for predecessors whose earliest starting date is earliest starting point 'E' - time task takes. And you go through recursively.

margins

It is the difference between earliest starting date of task and the latest (not delaying project).

free margin

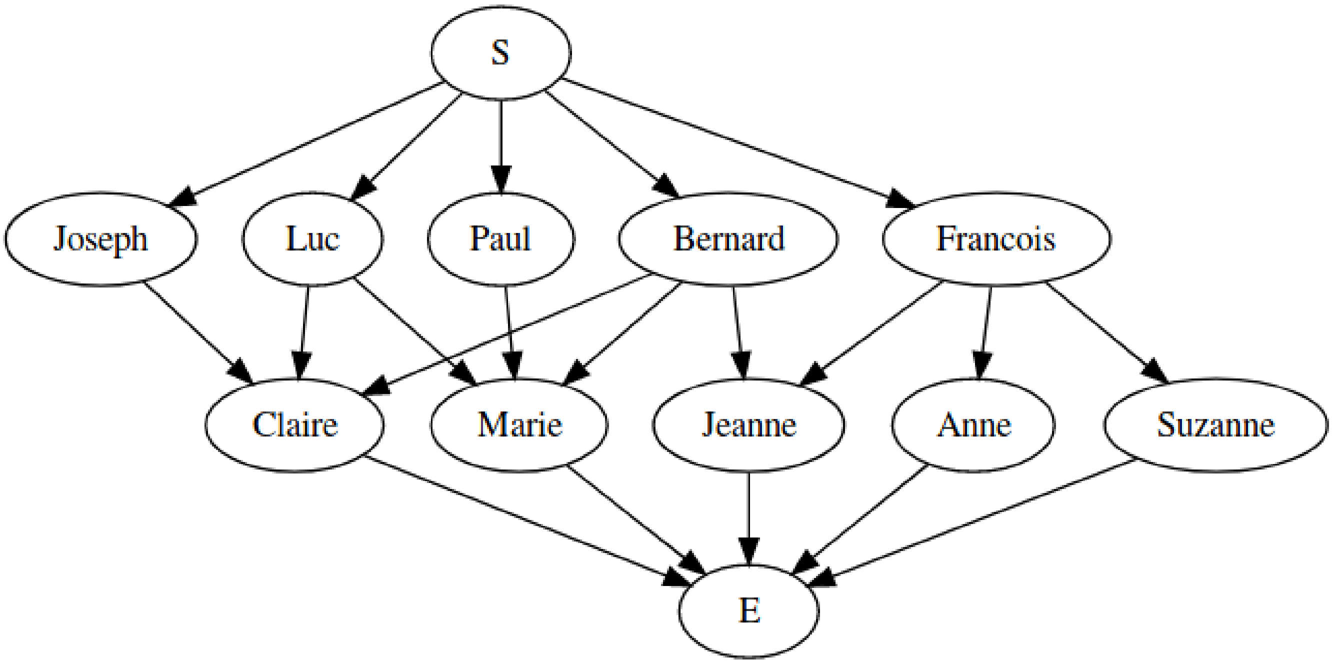
Is the difference between earliest starting date & latest one not modifying the earliest starting date of any successors.

2.3 A bit of formalism

Let $c : R_+$ be the capacity in a given graph G (having s anf t)
Then a flow on G is a function $f : A \rightarrow R_+$
Satisfying : (1) $\forall a \in A m f(a) <= c(a)$
(2) $\forall i \in V | \{s, t\}$:
$\sum f(a) = \sum f(a)$
$a \rightarrow v \qquad v \rightarrow a$
$a \in A \qquad a \in A$

2.4 Exercice 6 (exercise sheet)

1)



2) Simply apply "improving path" algo and you’re done.