

Algorithmes de flots

Quentin Fortier

October 11, 2021

Choix des chemins

On va voir plusieurs façon de trouver un chemin de s à t :

On va voir plusieurs façon de trouver un chemin de s à t :

- Parcours en profondeur
- Parcours en largeur
- Plus court chemin (Dijkstra, Bellman-Ford)
- Plus large chemin

Exercice

Écrire une fonction Python `ford_fulkerson` telle que `ford_fulkerson(G, s, t, path)` renvoie la valeur maximum d'un flot, où :

- `G` est un graphe orienté avec une capacité sur les arcs.
- `path` est une fonction qui à un graphe résiduel associe un chemin de `s` à `t`

Exercice

Écrire une fonction Python `ford_fulkerson` telle que `ford_fulkerson(G, s, t, path)` renvoie la valeur maximum d'un flot, où :

- `G` est un graphe orienté avec une capacité sur les arcs.
- `path` est une fonction qui à un graphe résiduel associe un chemin de `s` à `t`

Exercice

Implémenter le parcours en profondeur et l'utiliser avec `ford_fulkerson`.

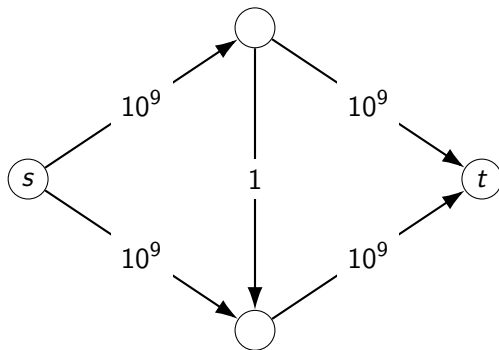
Quelle est la complexité de Ford-Fulkerson avec recherche des chemins par parcours en profondeur ? $O(|\vec{E}| \times |f^*|)$

Choix des chemins : Parcours en profondeur

La complexité $O(|\vec{E}| \times |f^*|)$ est atteinte pour ce genre de graphe :

Choix des chemins : Parcours en profondeur

La complexité $O(|\vec{E}| \times |f^*|)$ est atteinte pour ce genre de graphe :



Exercice

Implémenter le parcours en profondeur et l'utiliser avec `ford_fulkerson`.

Quelle est la complexité de Ford-Fulkerson avec recherche des chemins par parcours en profondeur ? $O(|\vec{E}| \times |f^*|)$