

Graphes et planification

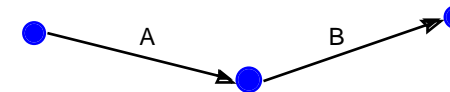
Un problème de planification
étant donné un travail à effectuer
nécessitant un (grand) nombre de tâches coordonnées entre elles

Trouver le temps minimum pour terminer le travail
Trouver les tâches critiques

Modèle PERT

Les tâches sont représentées par des arcs
La durée de la tâche est associée à l'arc
Les noeuds représentent des instants (début - fin de tâches)
2 instants particuliers : début travail - fin travail

Représentation de la dépendance "terminer A avant de commencer B"



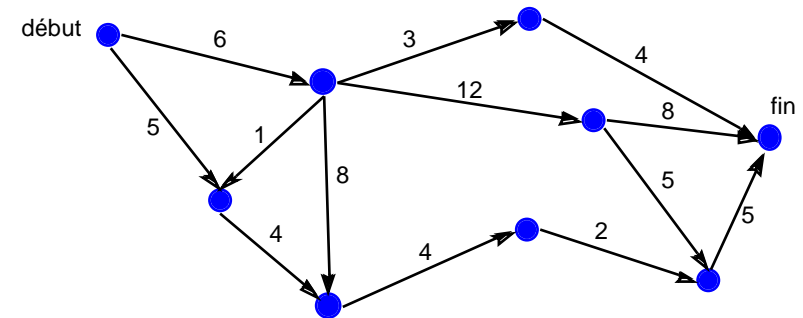
1. Vérifier que le graphe est sans cycle
2. Durée minimum du travail =
somme des durées le long du **plus long chemin** de début à fin.

Algorithme: trouver le chemin le plus long

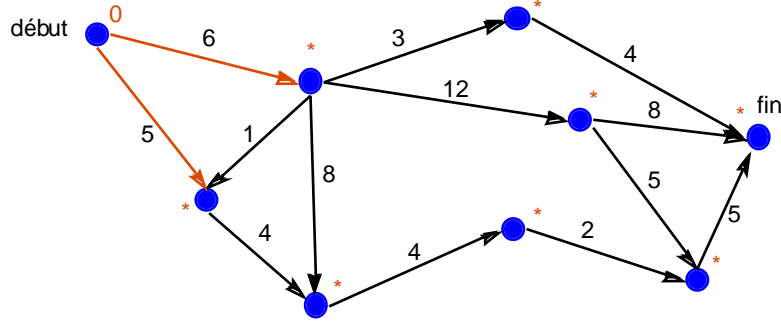
Algorithme "glouton"

```
Pour chaque noeuds x, fixer x.max = 0
fini = faux
Tant que (non fini) {
  fini = vrai
  Pour chaque arc a du graphe {
    si (a.origine.max + a.longueur > a.destination.max) {
      a.destination.max = a.origine.max + a.longueur; fini = faux;
    }
  }
}
```

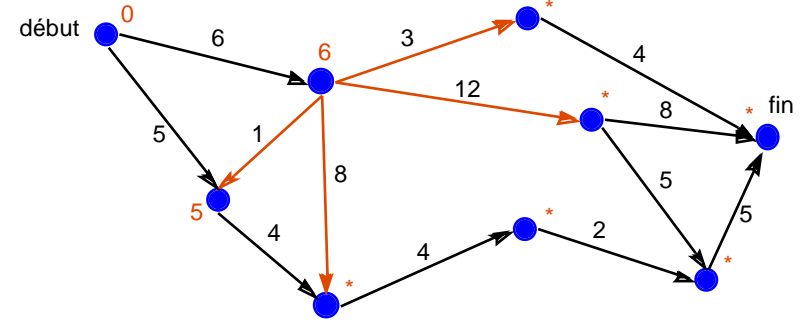
Exemple



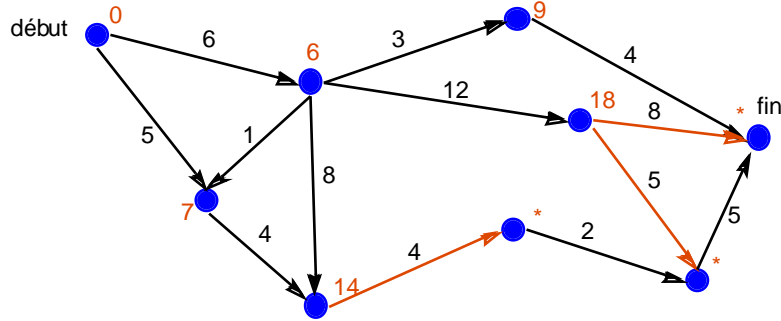
Trouver le chemin le plus long



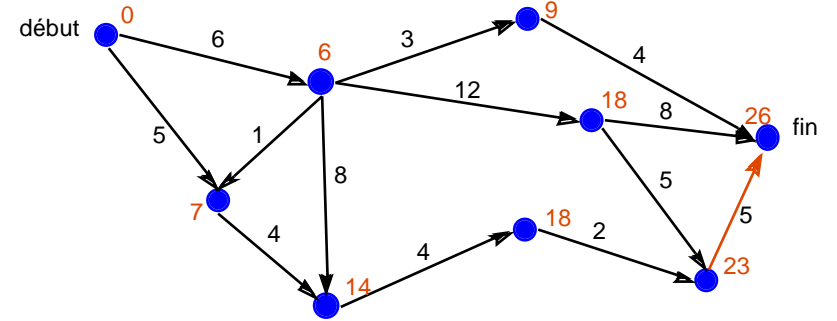
Trouver le chemin le plus long



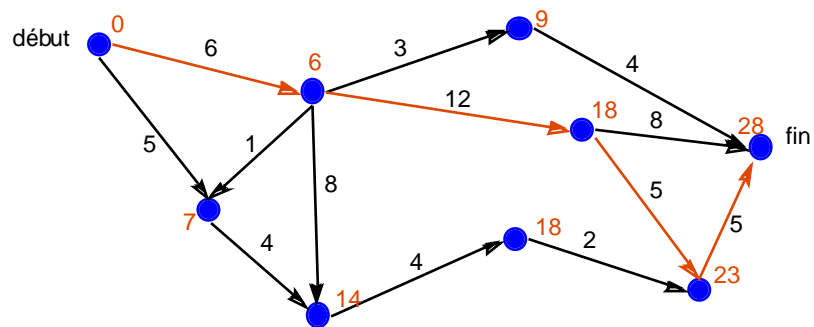
Trouver le chemin le plus long



Trouver le chemin le plus long



Chemin le plus long



Autre algorithme

Définition.

Un noeud est de rang R si

- tous ses arcs entrants partent de noeuds de rang $< R$.
- au moins un arc entrant part d'un noeud de rang $R-1$.

Le noeud de départ est de rang 0.



La notion de rang est bien définie s'il n'y a pas de circuit.

(preuve par récurrence)

Algorithme

Propriété

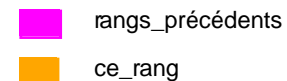
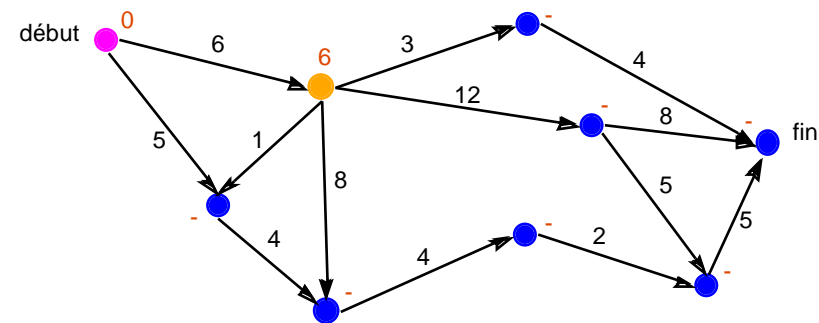
Tous les chemins du noeud de départ à un noeud de rang R ont au plus R arcs.

Algorithme

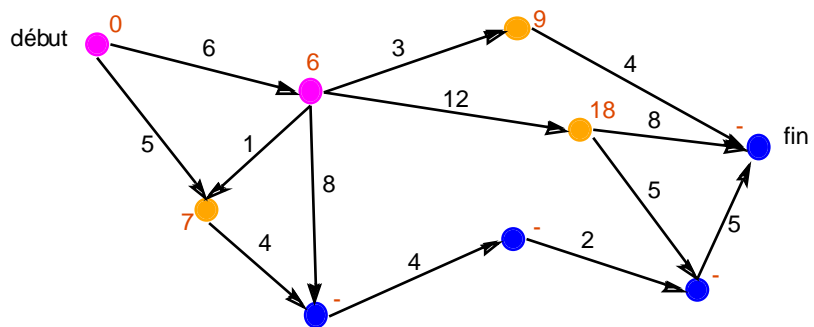
```

rangs_précédents ← {départ}
tant que arrivée non atteinte {
  ce_rang ← {}
  pour chaque noeud s ∉ rangs_précédents { // possible optimisation
    si tous les arcs entrants de s proviennent de rangs_précédents {
      ajouter s à ce_rang;
      s.max = MAXa entrant de s { a.origine.max + a.longueur }
    }
  }
  ajouter ce_rang à rangs_précédents;
}
    
```

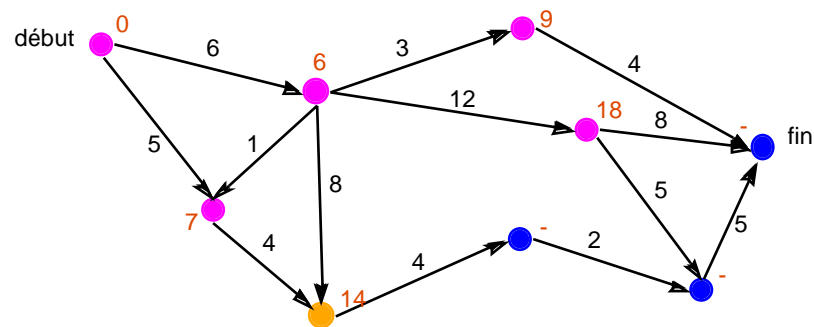
Trouver le chemin le plus long (algo 2)



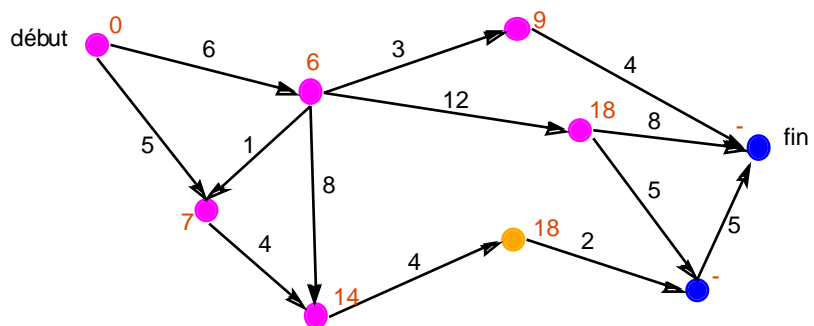
Trouver le chemin le plus long (algo 2)



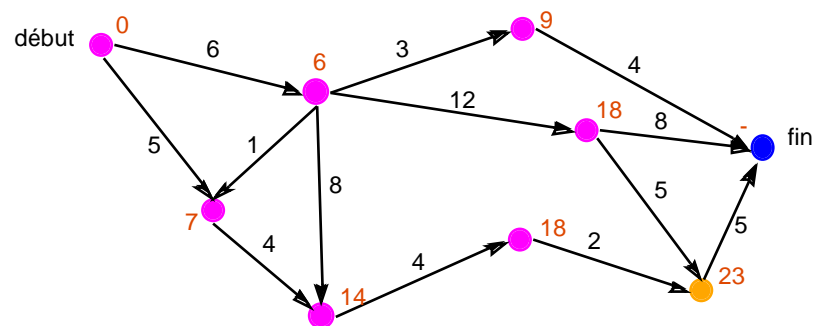
Trouver le chemin le plus long (algo 2)



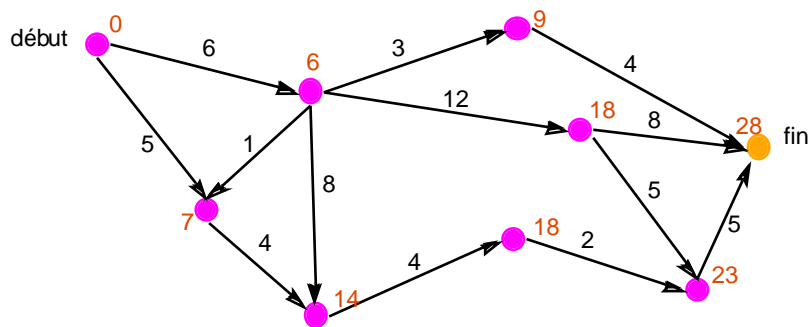
Trouver le chemin le plus long (algo 2)



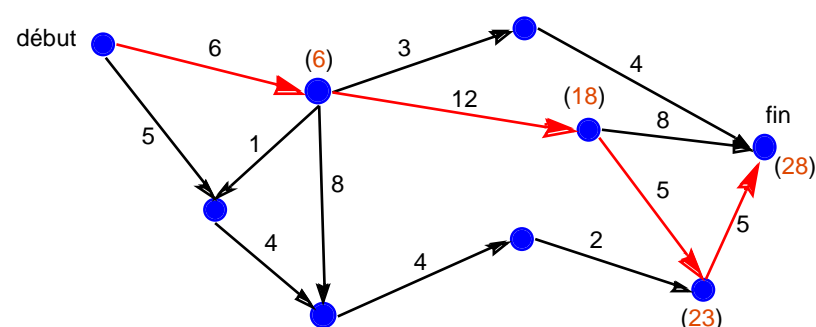
Trouver le chemin le plus long (algo 2)



Trouver le chemin le plus long (algo 2)

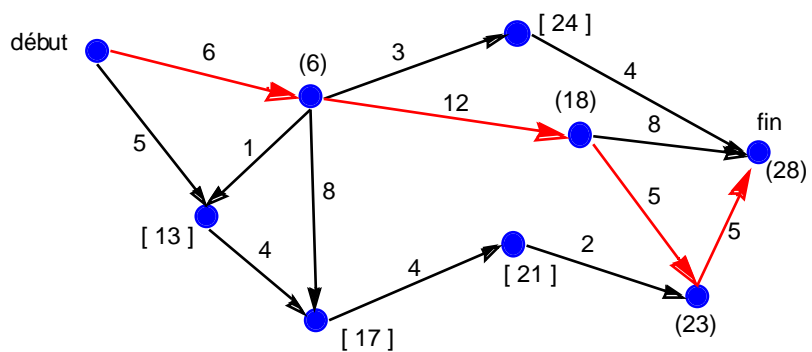


Chemin critique



=> temps de réalisation minimum = 28

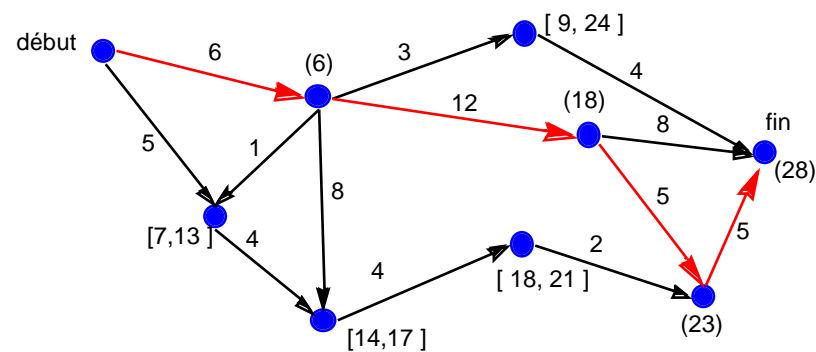
Dates "au plus tard"



Date limite de réalisation de chaque tâche.

Algorithme: parcours "en arrière" depuis la fin

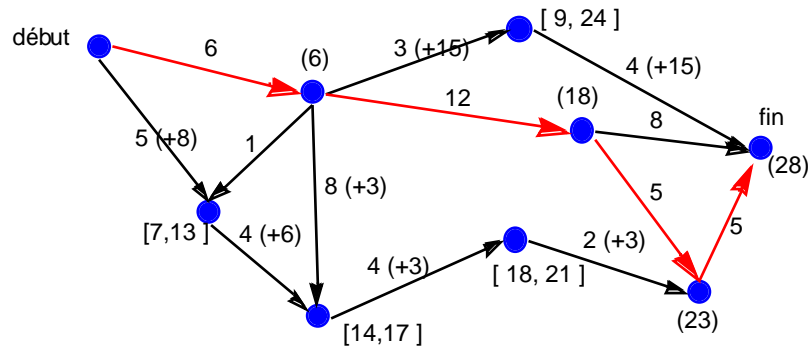
Dates "au plus tôt, au plus tard"



Analyse des marges sur les tâches

==> délai supplémentaire que l'on peut accorder à une tâche si les autres sont sans délai.

Marges



$\text{marge}(a) = a.\text{destination.au_plus_tard} - a.\text{origine.au_plus_tôt} - a.\text{durée}$

==> délai supplémentaire que l'on peut accorder à une tâche si les autres sont sans délai.

Graphes et connaissances

Utilisation des graphes comme modèle de représentation des connaissances

Réseaux sémantiques

Graphes conceptuels

Logique descriptive

Réseaux sémantiques

Noeuds du graphe : concepts ou objets

Arcs : liens sémantiques (type de lien)

Liens typiques :

"est un" (isa)

- "type de" (sous-classe, généralisation/spécialisation)

- "instance de" (appartenance à une classe)

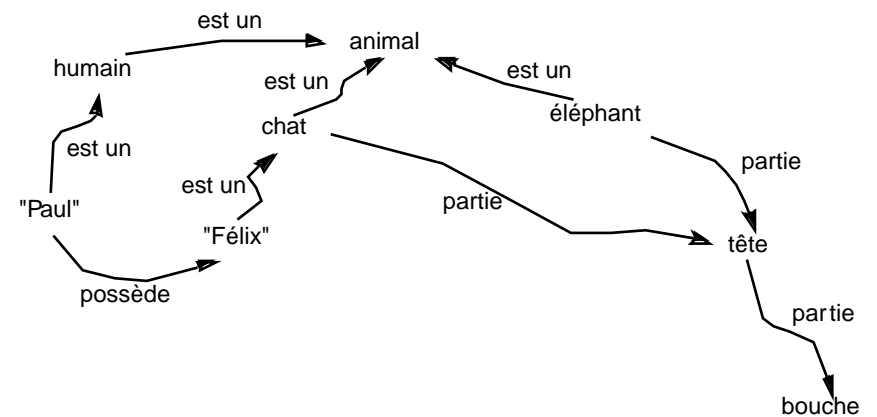
"est une partie de"

"propriété"

"conséquence"

(types de liens en fonction du but recherché)

Exemple

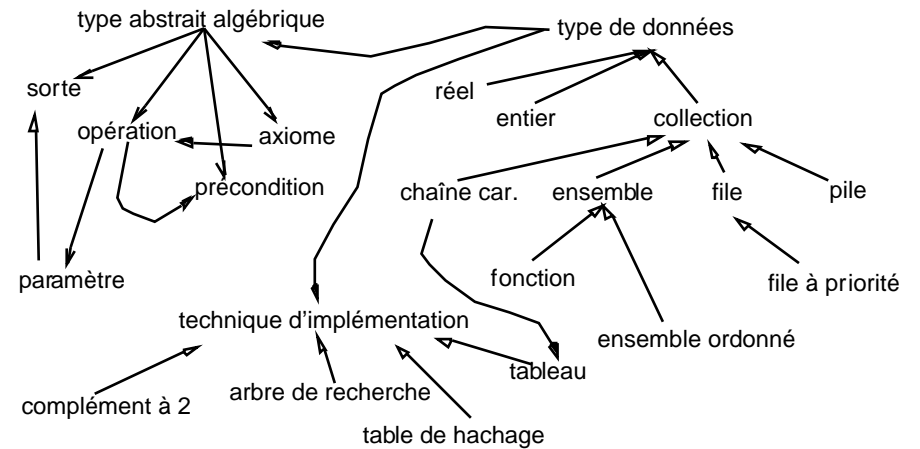


Calculs dans les réseaux sémantiques

Inférence de propriétés

- par composition (transitivité)
"Félix" (est-un) chat (est-un) animal --> "Félix" (est-un) animal
chat (partie) tête (partie) bouche --> chat (partie) bouche
- par héritage (est-un)
"Félix" (est-un) chat (partie) tête --> "Félix" (partie) tête
- par généralisation
"Paul" (possède) "Félix" (est un) chat --> "Paul" (possède) chat

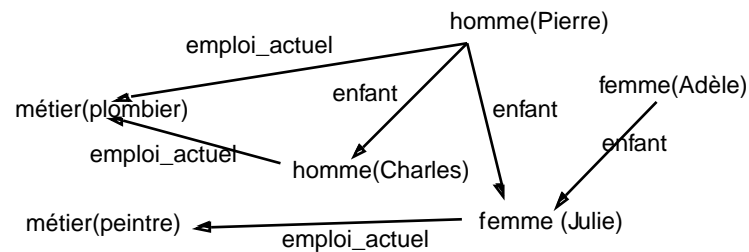
TP: les réseaux sémantique du cours



Logique descriptive

"schéma" : définitions de concepts, contraintes

"instances" : objets et rôles, chaque objet appartient à un concept



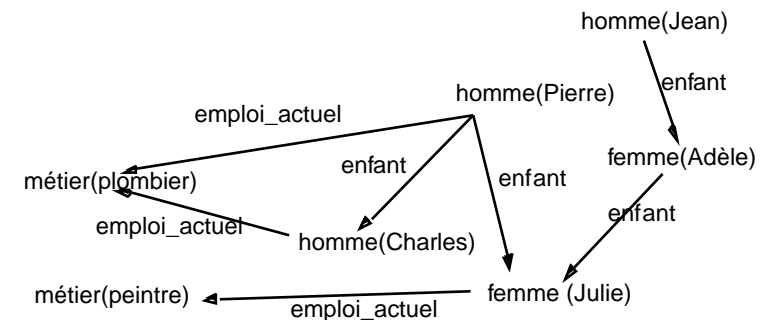
Définition de concepts

personne = homme \cup femme{Pierre, Charles, Julie, Adèle, Jean}

parent = \exists enfant:personne {Pierre, Adèle, Jean}

travailleur = \exists emploi_actuel : T{Pierre, Charles, Julie}

parent_sans_garçon = parent \cap \forall enfant:femme.....{Jean, Adèle}



Définition graphique

