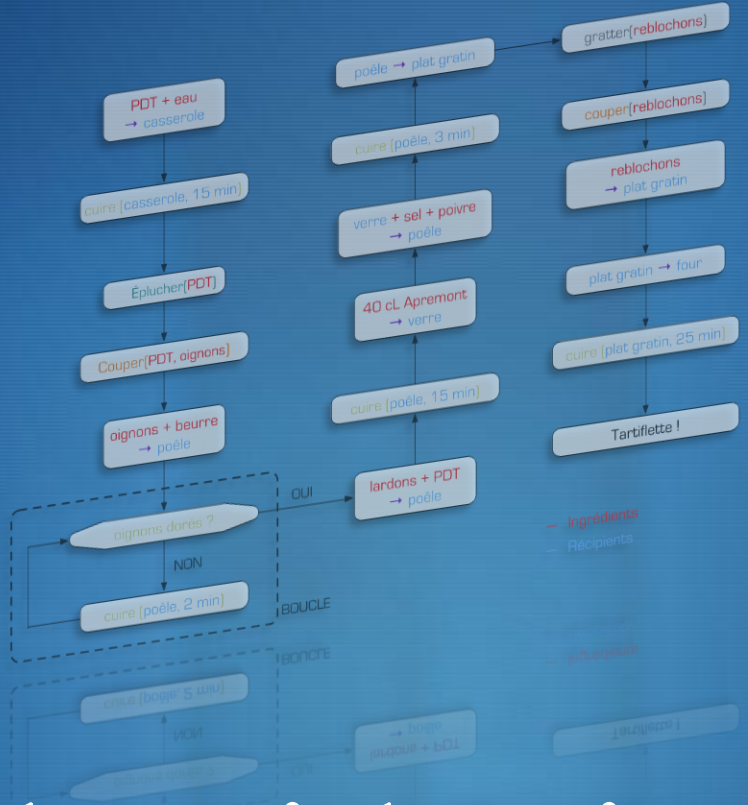



# Algorithmique

à. Introduction

G. GEVRIN

# SOMMAIRE

- Algorithme : définition
- Algorithmique
- Instructions
- Variables

# Tartiflette

Plat principal
Facile :
Moyen :

**Temps de préparation** : 30 minutes
**Temps de cuisson** : 20 minutes

**Ingrédients** (pour 6 personnes) :

- 1 reblochon et demi (2 pour les gourmands)
- 1,2 kg de pommes de terre
- 1 tranche de jambon fumé coupée en dés (ou bien des lardons ou des dés de bacon)
- 500 g d'oignons
- 40 cl d'Apremont ou autre blanc de Savoie sec (facultatif, mais donne plus de goût)
- 1 pincée de sel
- 1 pincée de poivre

**Préparation de la recette** :

Faites cuire les pommes de terre avec leur peau. Épluchez-les et coupez-les en rondelles.

Émincez les oignons puis faites-les revenir avec un peu de beurre. Une fois dorés, y ajouter le jambon (ou les lardons), ainsi que les pommes de terre. Laissez mijoter 15 min.

Si vous utilisez du vin blanc, c'est le moment de l'ajouter. Salez un peu, poivrez, laissez les pommes de terre s'imprégner du vin blanc quelques minutes avant de transférer le tout dans un plat à gratin.

Grattez au couteau les reblochons, coupez-les en 2 dans le sens de l'épaisseur et posez les sur les pommes de terre.

Faites cuire à four chaud (220°C) pendant 20 à 30 min. Servez avec une salade verte, voire quelques tomates, juste assaisonnées d'un peu de vinaigre d'échalote.

Recette proposée par Christophe_de_Marmiton

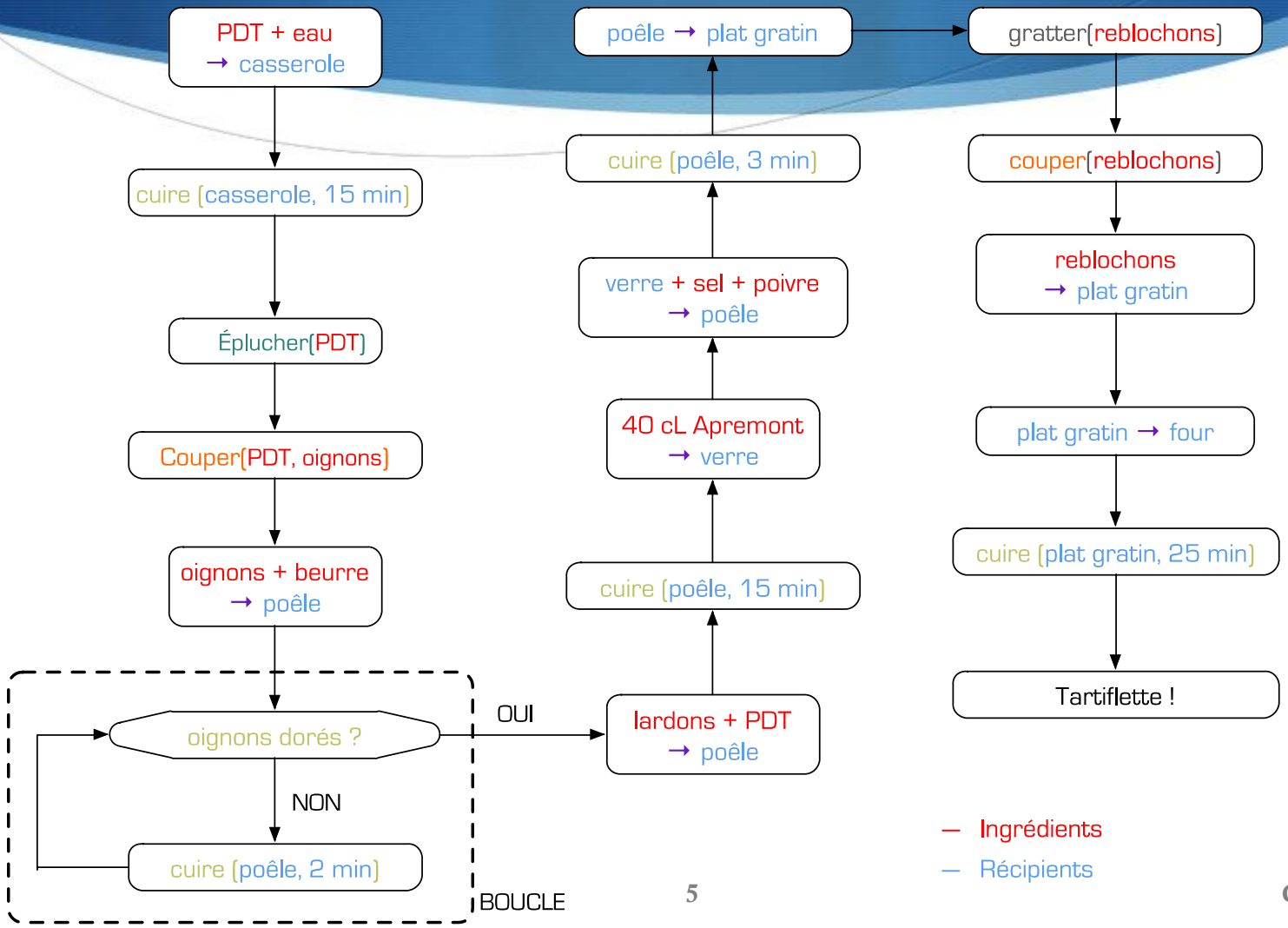
# Algorithme tartiflette

- Ingrédients :
  - 2 reblochons, 1,2 kg PDT, lardons, 500g oignons, 40 cL Apremont, sel, poivre, beurre, eau, salade (?)
- Récipients :
  - verre mesureur, poêle, casserole, plat à gratin, four
- Opérations de base :
  - mettre dans un récipient, mesurer, éplucher, couper, laisser cuire pendant X minutes, gratter, transférer contenu de … dans …

# Algorithme tartiflette

- Mettre 1,2 kg PDT dans casserole
- Mettre eau dans casserole
- Laisser cuire pendant 15 minutes
- Éplucher PDT
- Couper PDT
- Couper oignons
- Mettre beurre dans poêle
- Mettre oignons dans poêle
- Laisser cuire jusqu'à ce que oignons soient dorés
- Mettre lardons et PDT dans poêle
- Laisser cuire pendant 15 minutes
- Mettre 40 cL vin blanc dans verre mesureur
- Mettre vin blanc et sel et poivre dans poêle
- Laisser cuire pendant 3 minutes
- Transférer contenu de poêle dans plat à gratin
- Gratter reblochons
- Couper reblochons
- Mettre reblochons dans plat à gratin
- Mettre plat à gratin dans four
- Laisser cuire pendant 25 minutes (220°C)

# Organigramme tartiflette

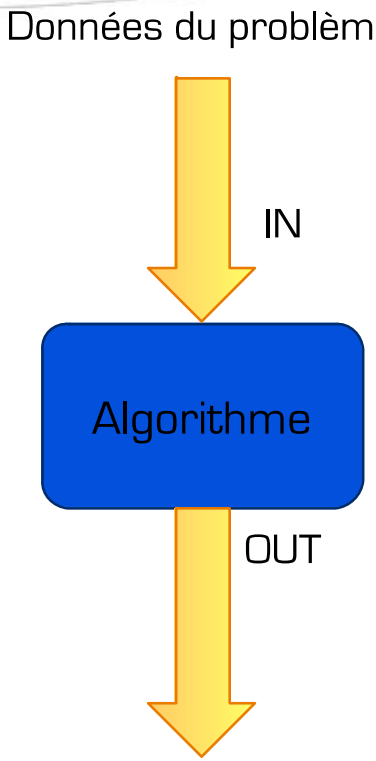


# A quoi ça sert ?

- À décrire les étapes de résolution d'un problème :
  - de façon structurée et compacte
  - à partir d'opérations de base
  - indépendamment d'un langage de programmation

- Etapes aussi appelées « pas de l'algorithme »

# A quoi ça sert ?



# A quoi ça sert ?

- À décrire les étapes de résolution d'un problème :
  - de façon structurée et compacte

→ Facile à comprendre, facile à transmettre

# A quoi ça sert ?

- À décrire les étapes de résolution d'un problème :
  - à partir d'opérations de base

→ Méthode de résolution adaptée :

- aux moyens à disposition
- aux connaissances de celui qui l'utilise

# A quoi ça sert ?

- À décrire les étapes de résolution d'un problème :
  - indépendamment d'un langage de programmation

→ Méthode de résolution :

- adaptée pour des problèmes qui se traitent sans ordinateur
- compréhensible sans apprendre un langage de programmation

# A quoi ça sert ?

- À décrire les étapes de résolution d'un problème :
  - indépendamment d'un langage de programmation

- Algorithmes antérieurs à l'informatique :
  - Euclide (vers -300) : calcul du PGCD de 2 nombres

  - Al-Khuwârizmî (825) : résolution d'équations

# Algorithmique - enjeux

- Algorithme **correct**
  - donne le résultat attendu ? ⟶ preuve de correction
  - quel que soit le type d'entrées ? ⟶ débuggage, tests unitaires
- Algorithme **rapide**
  - se termine ? ⟶ preuve de terminaison
  - en combien de temps ? ⟶ complexité

# Algorithmique - correction

- Correction : L'algorithme donne-t-il le résultat attendu ?
- Preuve de correction :
  - « invariant » : propriété vraie tout au long de l'algorithme :
    - vraie à la première étape
    - si vraie à une étape, vraie à l'étape suivante (démonstration par récurrence)
    - → vrai à la fin
- En pratique, pour débuter :
  - vérifier sur les "cas de base"
  - vérifier aux limites
  - vérifier sur des exemples aléatoires

# Algorithmique - terminaison

- Terminaison : L'algorithme se termine-t-il en un temps fini ?
- Preuve mathématique :
  - Démontrer que boucles rencontrent leur condition de terminaison
  - Exemple tartiflette : les oignons finiront bien par être dorés

# Algorithmique - complexité

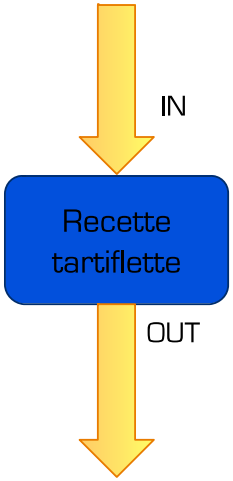
- Complexité : Combien de temps l'algorithme prend-il pour se terminer ?
- Théorie de la complexité :
  - En temps et en espace
  - nombre d'opérations en fonction de la taille du problème, dans le pire cas
  - prouver qu'on ne peut pas utiliser moins d'opérations pour résoudre le problème, dans le pire cas

# Algorithme - composants

- Entrée : données du problème
- Algorithme : instructions
- Sortie : résultat

# Algorithme - composants

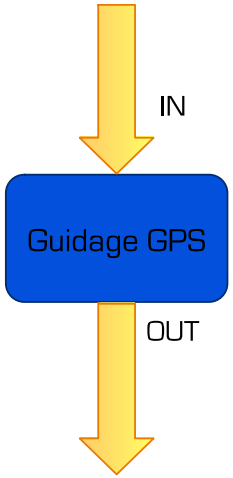
Ingrédients




Tartiflette

Infos sur l'environnement :

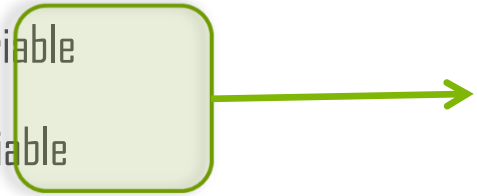
- position véhicule
- position destination
- cartes




Itinéraire

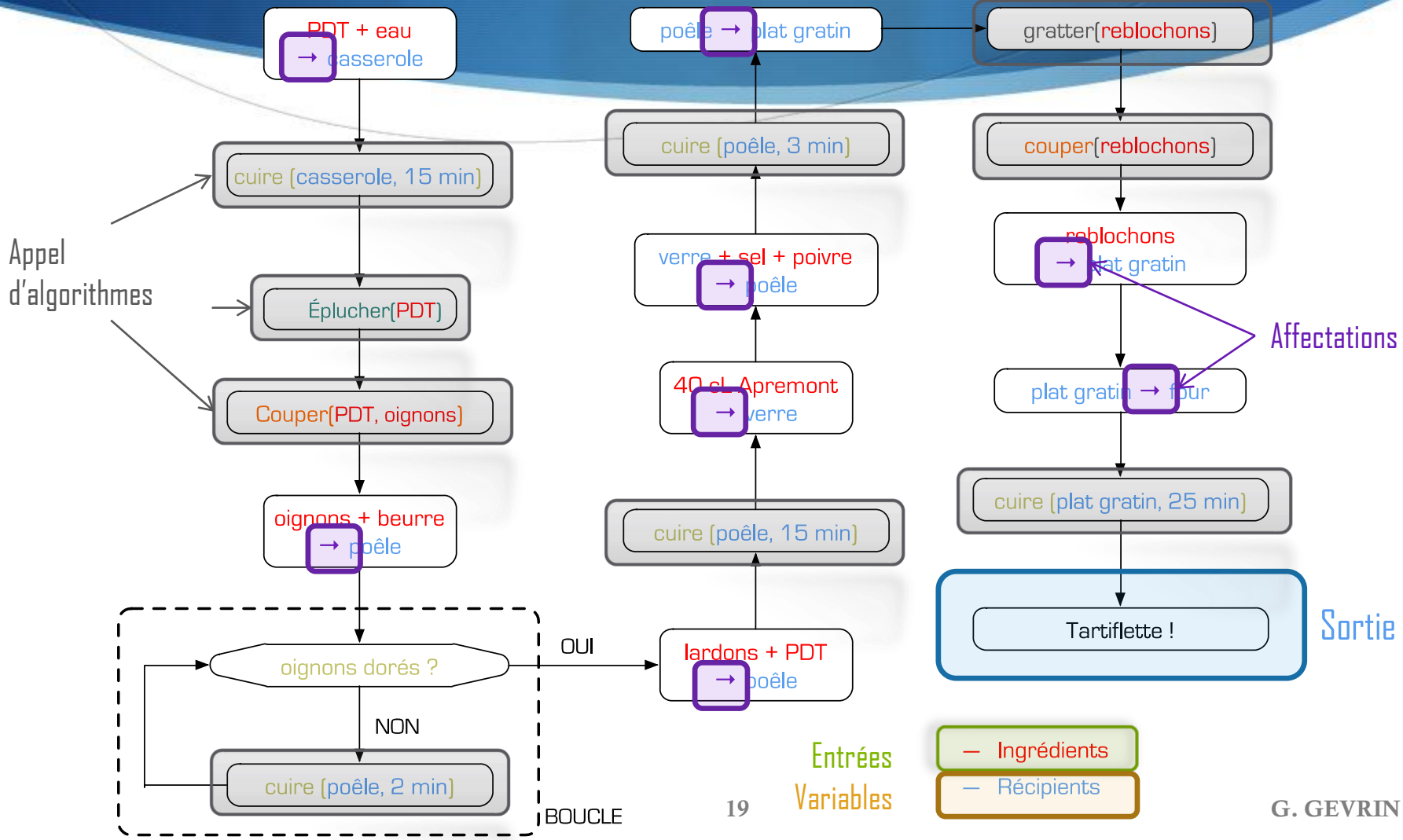
# Algorithme - instructions

- déclaration d'un algorithme
- appel d'un algorithme
- déclaration d'une variable
- affectation d'une variable
- entrées/sorties
- Boucle
- test



Stocker des valeurs, des résultats intermédiaires

# Organigramme tartiflette



# Variables et affectation

- Dans un algorithme, une variable possède :
  - un identifiant,
  - une valeur,
  - un type (ensemble des valeurs que peut prendre la variable).
- La valeur d'une variable :
  - est fixe à un moment donné,
  - peut changer au cours du temps.
- **Nom et type d'une variable ne changent pas** (même en cas de métamorphisme)

# Variables et affectation

- L'affectation change la valeur d'une variable (pseudo-code) :

  `a ← 5`

  - - la variable a prend la valeur 5
  - la valeur précédente est perdue ("écrasée")

  `a ← b`

  - la variable a prend la valeur de la variable b
  - la valeur précédente de a est perdue ("écrasée")
  - la valeur de b n'est pas modifiée
  - a et b doivent être de même type (ou de type compatible)

# Nommage des variables

- Dans un algorithme, choisir pour les variables :
  - un nom composé de lettres et éventuellement de chiffres
  - un nom expressif, par exemple :
    - chaine, requêtel… pour une chaîne de caractères
    - n, a, b, compteur, nbOperations, longueur… pour un entier
    - x, y, température pour un réel
    - estEntier, testEntier, trouvé… pour un booléen
  - un nom assez court (il faut l'écrire !), qui commence par une minuscule
  - éviter les noms réservés : pour, tant que, si…

# Nommage des variables

- Dans un programme :
  - éviter les lettres accentuées et la ponctuation
  - préférer l'anglais si votre code source est diffusé largement
  - être expressif et lisible :
  - Is_integer ou isInteger (camelCase) plutôt que isinteger