

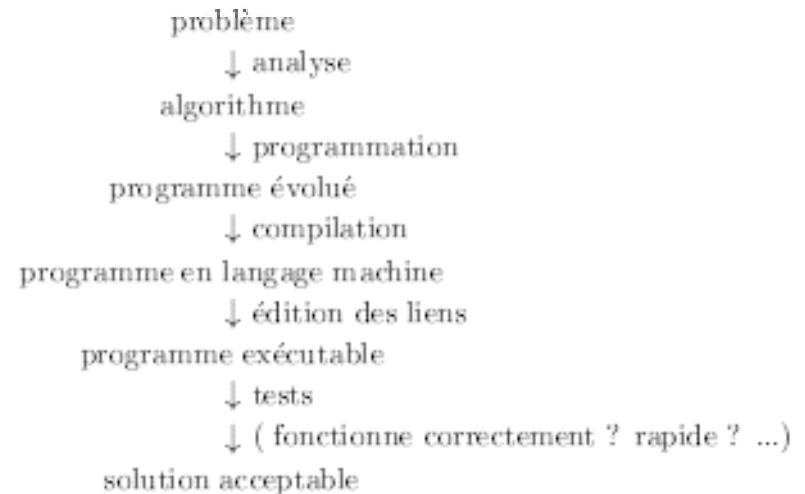
Analyse d'Algorithmes & Complexité

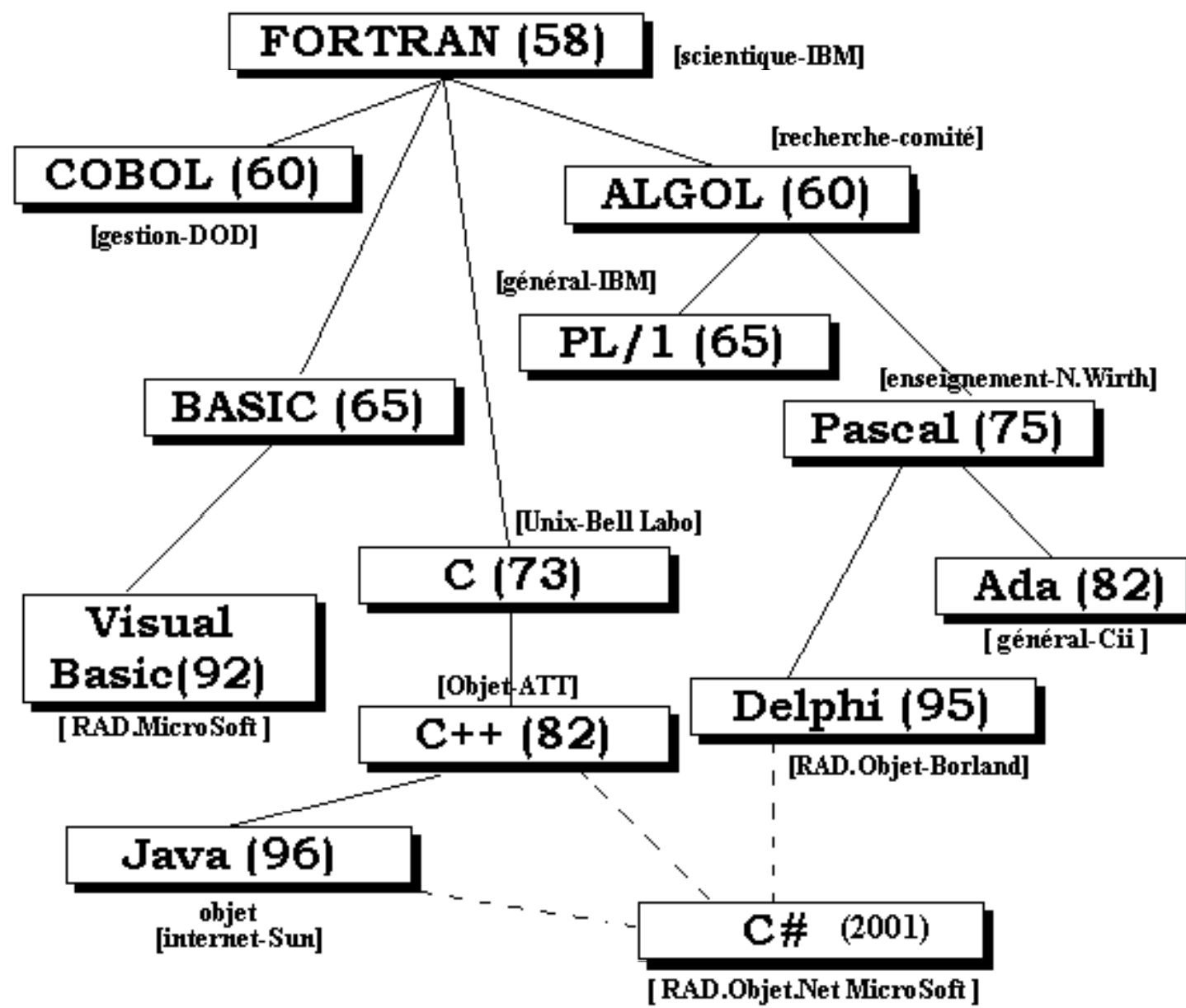
Qu'est ce qu'un Algorithme ?

Un **algorithme** est une "spécification d'un schéma de calcul sous forme d'une suite finie d'opérations élémentaires obéissant à un enchaînement déterminé", ou encore : la description des étapes à suivre pour réaliser un travail.

Un **programme** est (donc) la description d'un algorithme dans un langage « **évolué** » accepté par la machine.

Un algorithme, à l'inverse d'un **programme**, est **indépendant** du langage de programmation (et donc de la machine).

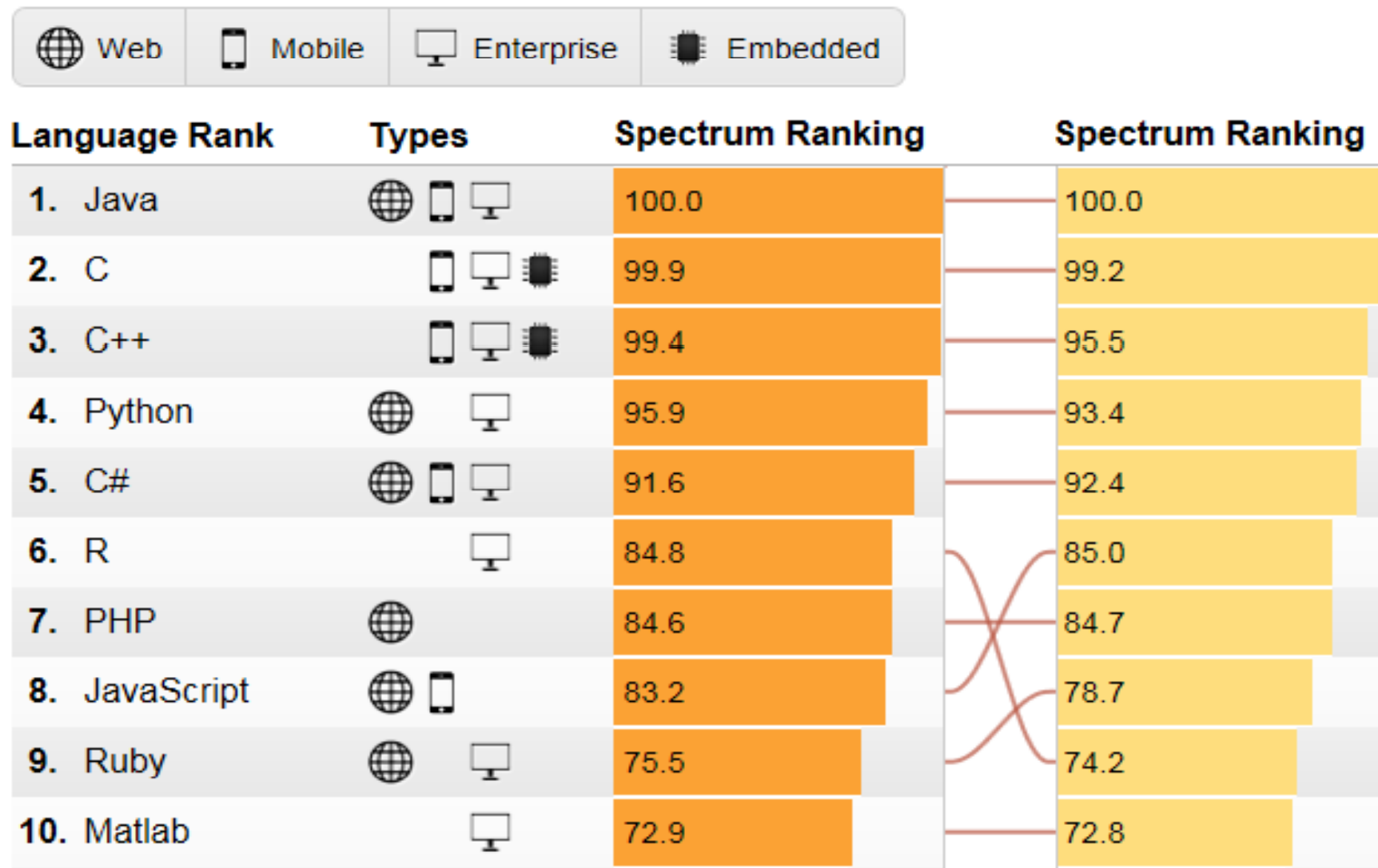




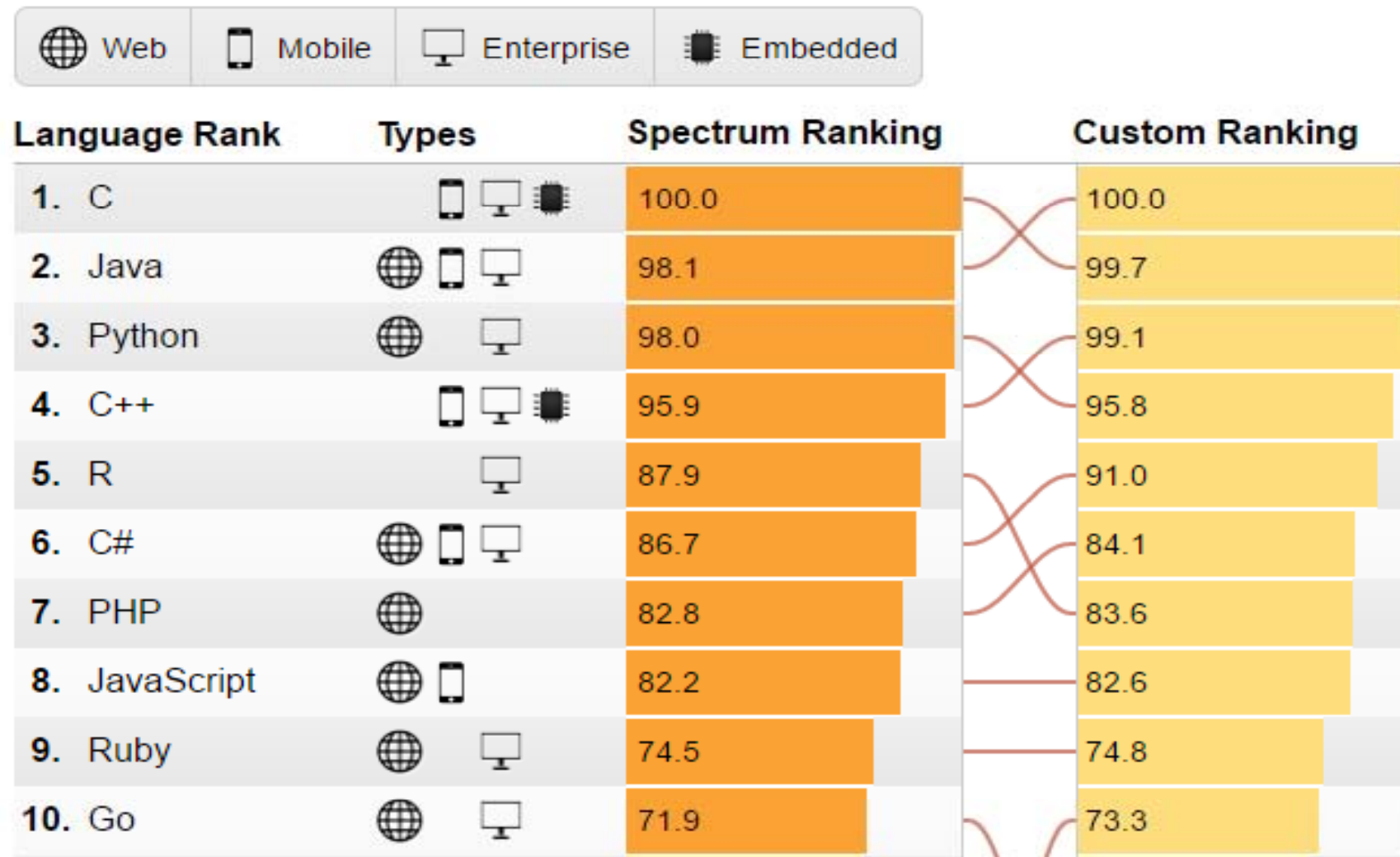
Langage de programmation :

- **ADA**: Ada Lovelace, qui est supposée avoir écrit le premier programme de l'histoire.
- **PLI** : Programming Language number 1
- **Cobol** : COmmon Business Oriented Language
- **Algol** : algorithmic oriented language
- **Prolog** : PROgrammation LOGique (IA)
- **LISP** : list processing (IA)
- **RAD** : "Rapid Application Development". Un langage RAD permet de développer des applications en très peu de temps dans un environnement souvent visuel. Delphi est un exemple de RAD.

Meilleurs langages de programmation en 2015 :



Meilleurs langages de programmation en 2016 :



Paradigmes:

Un *paradigme* est un style de programmation.

- **Impératif** : ou *procédural* est basé sur l'idée d'une exécution étape par étape semblable à une recette de cuisine.
- **Déclaratif**: Les deux paradigmes déclaratifs sont: fonctionnel et logique. En paradigme fonctionnel le programmeur décrit des fonctions mathématiques. En paradigme logique il décrit des prédicats.
- **Fonctionnel**: est basé sur l'idée d'évaluer une formule, et d'utiliser le résultat pour autre chose
- **Logique**: est basé sur l'idée de répondre à une question par des recherches sur un ensemble, en utilisant des axiomes, des demandes et des règles de déduction.

Paradigmes:

- **Orienté objet:** est destiné à faciliter le découpage d'un grand programme en plusieurs modules isolés les uns des autres.
- **Concurrent:** un programme peut effectuer plusieurs tâches en même temps.
- **Visuel:** le but de faciliter la programmation des interfaces graphiques.
- **Événementiel:** le programme n'attend rien et est exécuté lorsque quelque chose s'est passé
- **Basé web:** Java, PHP et Javascript sont des langages de programmation basée web

Conception d'un algorithme :

- **Analyse descendante** : décomposer le problème en sous problème de plus petite taille jusqu'à aboutir aux instructions élémentaires.
- **Analyse ascendante ou montante** : utiliser des fonctions, des primitives et des outils dont on dispose, les assembler pour faire un truc qui résout notre problème.
- **Mélange des deux (meilleure)** : on fait une analyse descendante tout en ayant à l'esprit les modules bien conçus.

Qualités d'un algorithme

1. **Qualité d'écriture** : un algorithme doit être structuré, modulaire, avec des commentaires pertinents, etc. *Il faut pouvoir comprendre la structure d'un coup d'oeil rapide.*
2. **Terminaison** : le résultat doit être atteint en un nombre fini d'étapes. (Pas de boucles infinies, étudier tous les cas possibles. *Par exemple dans une boucle il faut montrer que la(es) variable(s) intervenant dans le test de la boucle décroi(ssen)t.*
3. **Validité** : le résultat doit répondre au problème demandé. *Attention, un jeu d'essais ne prouve **JAMAIS** qu'un programme est correct. Il peut seulement prouver qu'il est faux. .*
4. **Performance** : étude du coût (complexité) en temps et en mémoire.