artement de Mathématiques et Informatique
ulté des sciences et de la Technologie
ersité Djilali Bounâama Khemis Miliana

Algorithmique Avancée et Complexité

1ére année de Master Ingénierie du Logiciel

Par: Mr Haniche Fayçal

Problématique

- Exemple : Algorithme de calcule du n^{ème} membre de la suite de Fibnocci F_n

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{si } n>1, \\ 1 & \text{si } n=1, \\ 0 & \text{si } n=0. \end{cases}$$

La suite génère les nombres suivants : $0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$

Il existe l'approximation suivante: $F_n \approx 2^{0.694n}$

Problématique

Algorithm 1 FIB1(n)

```
1: if n = 0 then
2:     return 0
3: else if n = 1 then
4:     return 1
5: else
6:     return fib1(n − 1) + fib1(n − 2)
7: end if
```

Trois questions essentielles sont posées :

1. L'algorithme est-il correct ?
2. Combien de temps exige-t-il p terminer, en fonction de n ?
3. Peut-on l'améliorer ?

Problématique

- **Q1 :** l'algorithme est bien correct, car il met en œuvre exactement la définition récurrente de la suite de Fibnocci
- Q2 : Soit T(n) le nombre de pas dont a besoin l'algorithme pour calculer F_n

Une première évidence :

$$T(n) \leq 2 \text{ pour } n \leq 1.$$

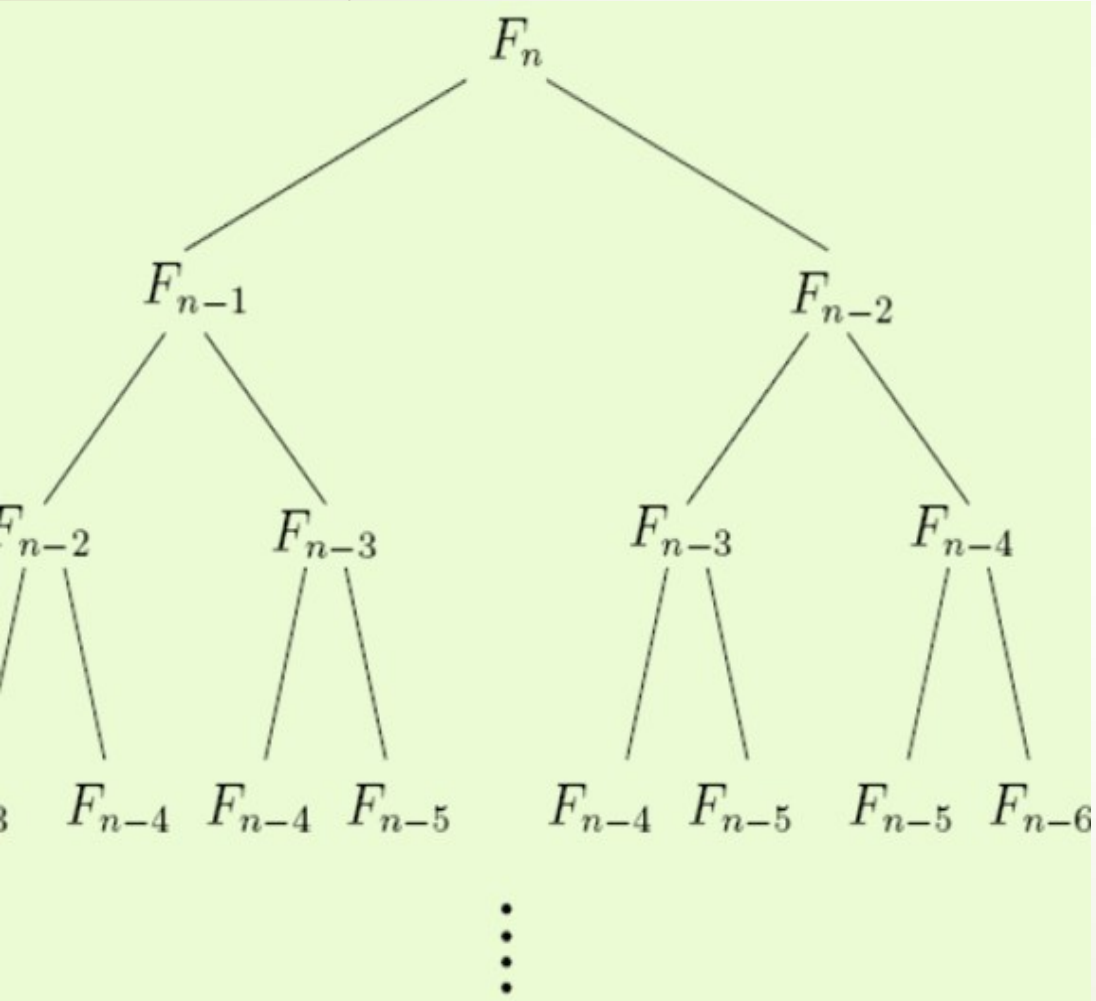
et Pour des nombres n plus grands,

$$T(n) = T(n-1) + T(n-2) + 3 \text{ pour } n > 1.$$

Ceci veut dire que le nombre de pas croit aussi vite que la suite de Fibnocci. Comme la suite croit exponentiellement, **T(n) a évidemment une croissance exponentielle.**

Problématique

- Explication: le comportement de l'algorithme fib1.



Soit par exemple le calcul de F_{200}

$$T(200) \geq F_{200} \geq 2^{138}$$ Pas élémentaire

Question: Temps nécessaire pour ce calcul sur un ordinateur ?

Exemple sur le NEC Earth Simulator, qui exécute 40 milliards ($40 * 10^{12}$) instructions par seconde.

Résultat : 2^{96} secondes = …….. Jours ?

Même si nous augmentant considérablemen capacité des Machine ,
le problème demeura toujours !!!!!

Objectif du cours

- Objectif 1 : Evaluation de la complexité des Algorithme en terme de temps de calcule.
- Objectif 2 : Etude de quelques techniques d'optimisation des Algorithmiques
- Objectif 3 : présentation des classes de problémes en terme de complexité

Plan du cours

1. Les bases de l'analyse des algorithmes
2. Analyse des algorithmes de recherche
3. Analyse des algorithmes de tri
4. Analyse des algorithmes récursifs
5. Les classes de problèmes
6. Algorithmique des arbres
7. Algorithmique du texte