

Atelier Algo n°1

Augustin, Thomas & François

A photograph of a prison corridor with metal bars. Several prisoners are visible, with their hands pressed against the bars. Some are holding white plastic bags. The corridor is dimly lit, with a small light source visible in the distance.

Bribe the Prisoners

Google Jam 2009

Du côté de Fleury...

- P cellules qui se suivent.
- Q prisonniers à libérer l'un après l'autre.
- Quand on libère un prisonnier, la **rumeur** se répand de cellule en cellule, jusqu'à tomber sur une cellule vide ou l'extrémité de la prison.
- Les prisonniers mis au courant commencent à **s'agiter**.
- Le maton paye chaque prisonnier **1 EURO** pour qu'il se calme.

**Étant donnée une liste de prisonniers à libérer,
quelle somme d'argent minimale est à
prévoir?**



GO GOOGLE!

[https://code.google.
com/codejam/contest/189252/dashboard#s=p2](https://code.google.com/codejam/contest/189252/dashboard#s=p2)

Programmation Dynamique

Deux propriétés à vérifier :

- ❑ “**Optimal Substructure**” : une solution optimale du problème peut être construite grâce aux **solutions optimales** de ses **sous-problèmes**.
- ❑ “**Overlapping sub-problems**” : le problème se décompose en sous-problèmes résolus **plusieurs fois** lors, par exemple, d’appels récursifs.

“Optimal Substructure” - Exemple

Plus court chemin :

Si le **plus court chemin d’Ambert à Paris** passe par **Dijon** et **Orléans**, le plus court chemin de **Dijon à Paris** passe aussi par **Orléans**.



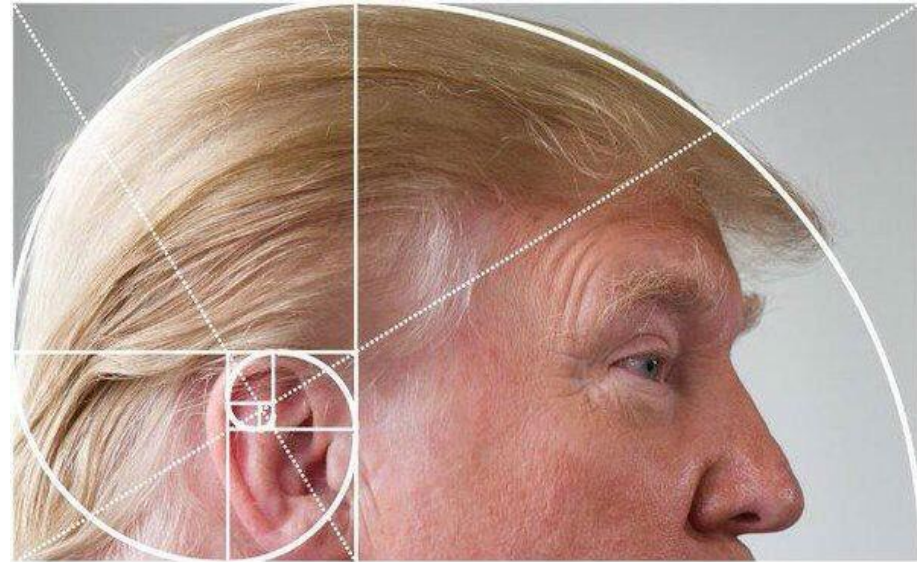
“Overlapping Sub-problems” - Exemple

Pour calculer **$F(n+2)$** il faut calculer une fois $F(n+1)$ et une fois $F(n)$.

Ou encore **deux fois $F(n)$** et une fois $F(n-1)$.

Suite de Fibonacci

- ❑ $F(0) = 1, F(1) = 1$
- ❑ $F(n+2) = F(n+1) + F(n)$



MERCI ;-)

