

Université Paris Diderot, Master 1 II

Projet d'algorithmique - étude et présentation d'un algorithme

Année universitaire 2010-2011

## Remarque

- { Si vous avez une mauvaise note du partiel (moins de 6) ne faites pas ce projet. Concentrez-vous sur le cours, les TDs, les annales, pour bien réussir l'examen.

## Déroulement du projet

- Ce projet s'effectue par binôme. Chaque binôme doit
- { Choisir un algorithme dans la liste proposée et s'inscrire via la page web <http://www.doodle.com/b6wgr794a2fxbz97>. Les premiers venus auront bien sûr un choix plus large.
  - { Vous pouvez également prendre un algorithme absent sur la liste. Votre choix doit être valide par Eugene Asarin. Pour ceci envoyez-lui avant le 10/12/11 un mel avec objet [Algo M1 : projet libre] contenant l'information suivante :
    - { vos noms ;
    - { le nom de l'algorithme (ou du problème algorithmique) que vous souhaitez présenter ;
    - { un lien, une référence, ou un document sur cet algorithme.
  - { Etudier l'algorithme en utilisant la littérature et l'Internet.
  - { Préparer un exposé de 25' sur l'algorithme.
  - { S'inscrire sur le web pour une soutenance orale. Les inscriptions seront ouvertes en décembre.
  - { Présenter l'exposé lors d'une soutenance orale de 30' en janvier 2011 (après le 15). Vous aurez à votre disposition un vidéo-projecteur, et si besoin est, un PC portable. Il n'y a pas de rapport à rendre.

## Contenu technique

Vous devez identifier et présenter clairement la plupart des éléments suivants :

- { Le problème algorithmique
- { L'algorithme qui résout ce problème
- { Les grands principes utilisés dans cet algorithme
- { Un exemple
- { La preuve de correction
- { Détails d'implémentation : espace, choix de structures de données
- { Analyse de complexité
- { Vos sources d'informations.

## Quelques conseils

- { Avant de choisir définitivement un algorithme trouvez-le sur le web pour évaluer sa difficulté et son intérêt.
- { Après avoir compris et étudié l'algorithme, prenez votre temps pour préparer la présentation.
- { Soyez très pédagogiques. Il est possible qu'un membre du jury ne connaît pas l'algorithme, il doit le comprendre en suivant votre exposé.

- { Il **n'est pas demandé** de programmer l'algorithme, vous pouvez le faire seulement si vous en avez envie.
- { Si vous souhaitez, vous pouvez donner dans votre expose autre information pertinente : histoire du probleme et de l'algorithme, applications etc. C'est conseille en cas ou votre algorithme est facile.
- { Si vous avez *vraiment* besoin d'un papier de recherche disponible sur un site payant d'editeur, contactez E. Asarin.

## Les algorithmes

1. Chemin le plus court pour tous les couples : algorithme de Johnson. *all pairs shortest path*
2. Chemin le plus court pour tous les couples : algorithme par carre de matrice en algebre min-plus. *all pairs shortest path*
3. Algorithme de tri Radix-sort.
4. Probleme 2SAT et un algorithme de solution.
5. Arbres 2-3-4 et algorithmes associes.
6. Arbres rouges-noirs et algorithmes associes.
7. Arbres Patricia et algorithmes associes. *Patricia trees*
8. Recherche de sous-chaine : algorithme de Rabin-Karp. *string-matching*
9. Distance de Levenshtein
10. Alignement de sequences : algorithme de Needleman-Wunsch *alignment*
11. Alignement de sequences : algorithme de Hirschberg *alignment*
12. Recherche de PGCD : algorithme binaire *binary GCD*.
13. Tableaux de suffixes et algorithmes associes. *suffix array*
14. Arbre de suffixes et algorithmes associes *suffix tree*.
15. Reduction transitive.
16. Tas binomial et son application a l'algorithme de Dijkstra.
17. Enveloppe convexe : algorithme de Graham. *convex hull*
18. Enveloppe convexe : algorithme de Jarvis. *convex hull*
19. Arbre couvrant minimum : algorithme de Boruvka *spanning tree*.
20. Union-find et son application a l'algorithme de Kruskal.
21. Code prefixe optimal : algorithme de Huffman.
22. Voyageur de commerce : methode *branch and bound*.