Probabilités et simulation

application à l'analyse d'algorithmes et à la randomization

Jean-Marc Vincent[1]

[1]Laboratoire ID-IMAG
MESCAL Project
Université Joseph Fourier, Grenoble
Jean-Marc.Vincent@imag.fr
http://www-id.imag.fr

Exemple : complexité d'algorithmes

n données à trier

- choix d'un pivot,
- séparation de l'ensemble des données plus grandes M et plus petites m
- trier récursivement M et m
- assembler les données triées

Complexité : au pire n^2, au mieux $\mathcal{O}(n \log n)$.
cas le pire, cas le meilleur ?
analyse en moyenne
$\Rightarrow$ **comprendre le comportement de l'algorithme**

Exemple : complexité d'algorithmes

n données à trier

- choix d'un pivot,
- séparation de l'ensemble des données plus grandes M et plus petites m
- trier récursivement M et m
- assembler les données triées

Complexité : au pire n^2, au mieux $\mathcal{O}(n \log n)$.

cas le pire, cas le meilleur ?

analyse en moyenne

$\Rightarrow$ **comprendre le comportement de l'algorithme**

Exemple : complexité d'algorithmes

n données à trier

- choix d'un pivot,
- séparation de l'ensemble des données plus grandes M et plus petites m
- trier récursivement M et m
- assembler les données triées

Complexité : au pire n^2, au mieux $\mathcal{O}(n \log n)$.

cas le pire, cas le meilleur ?

analyse en moyenne

$\Rightarrow$ **comprendre le comportement de l'algorithme**

Exemple : complexité d'algorithmes

n données à trier

- choix d'un pivot,
- séparation de l'ensemble des données plus grandes M et plus petites m
- trier récursivement M et m
- assembler les données triées

Complexité : au pire n^2, au mieux $\mathcal{O}(n \log n)$.

cas le pire, cas le meilleur ?

analyse en moyenne

$\Rightarrow$ **comprendre le comportement de l'algorithme**

Exemple : randomization d'algorithmes

n données à trier

- choix **aléatoire** d'un pivot,
- séparation de l'ensemble des données plus grandes M et plus petites m
- trier récursivement M et m
- assembler les données triées

Complexité : au pire n^2, au mieux $\mathcal{O}(n \log n)$.
cas le pire, cas le meilleur ?
$\Rightarrow$ **analyse en moyenne indépendante des données en entrée en** $\mathcal{O}(n \log n)$

Exemple : randomization d'algorithmes

n données à trier

- choix **aléatoire** d'un pivot,
- séparation de l'ensemble des données plus grandes M et plus petites m
- trier récursivement M et m
- assembler les données triées

Complexité : au pire n^2, au mieux $\mathcal{O}(n \log n)$.

cas le pire, cas le meilleur ?

$\Rightarrow$ **analyse en moyenne indépendante des données en entrée en** $\mathcal{O}(n \log n)$

Exemple : randomization d'algorithmes

n données à trier

- choix **aléatoire** d'un pivot,
- séparation de l'ensemble des données plus grandes M et plus petites m
- trier récursivement M et m
- assembler les données triées

Complexité : au pire n^2, au mieux $\mathcal{O}(n \log n)$.

cas le pire, cas le meilleur ?

$\Rightarrow$ **analyse en moyenne indépendante des données en entrée en** $\mathcal{O}(n \log n)$

Exemple : randomization d'algorithmes

n données à trier

choix **aléatoire** d'un pivot,
séparation de l'ensemble des données plus grandes M et plus petites m
trier récursivement M et m
assembler les données triées

Complexité : au pire n^2, au mieux $\mathcal{O}(n \log n)$.
cas le pire, cas le meilleur ?
$\Rightarrow$ **analyse en moyenne indépendante des données en entrée en** $\mathcal{O}(n \log n)$

Contrôle de l'aléatoire : contexte des réseaux

canal de communication bruité
taux de bruit $\Rightarrow$ contrôle d'erreur
qualification du protocole
ex : bit de parité, parité horizontale/verticale, CRC...
$\Rightarrow$ maîtriser l'environnement

Contrôle de l'aléatoire : contexte des circuits

composants électroniques
durée de vie $\Rightarrow$ mode de fonctionnement dégradé
duplication des fonctionnalités
dimmensionnement
$\Rightarrow$ maîtriser l'environnement

Algorithmes randomisés : contexte des réseaux

canal de communication partagé pas de synchronisation

répéter
écoute de la porteuse
si la porteuse est disponible **alors**
émission de l'entête
si pas de collision détectée **alors**
on transmet la totalité du message
sinon
on interrompt la transmission
fin si
fin si
on attend un **certain** temps avant de recommencer
jusqu'à transmission complète du message

Exemple de règle : à chaque échec on double l'intervalle I et on choisit uniformement sur I

$\Rightarrow$ **utiliser *RANDOM* pour mélanger les comportements.**

Algorithmes randomisés : contexte de la cryptographie

cryptage clé publique/privée basé sur la décomposition de grands entiers
avoir de grands nombres premiers
difficiles à générer
générer p avec une quasi-certitude qu'il est premier
algorithme de Miller-Rabin
$\Rightarrow$ **réduire la complexité d'un algorithme (acceptable).**

Objectif du cours

Objectif

1. Acquérir et maîtriser le langage des probabilités dans le contexte informatique (modélisation)
2. Savoir générer des données distribuées selon une loi donnée (écrire les algorithmes).
3. Savoir construire des plans d'expérience simples et savoir analyser les résultats avec rigueur.

Méthodes

1. Cours : aspects fondamentaux, grands exemples
2. TD : exercices avec résumé du cours, écriture d'algorithmes et de preuves, mise en œuvre
3. Expérimentation dans le cadre d'un mini-projet (éventuellement 2)

Evaluation

1. Examen écrit
2. Contrôle continu : Mini-projets + Quicks

Plan du cours

Partie I : Structures discrètes

1. Analyse de données expérimentales
   TD : Le générateur Random
2. Variables aléatoires de loi discrète
   TD : Transformation de générateurs
3. Génération selon une loi discrète donnée
   TD : Génération de loi discrète : distribution classique / arbitraire / non borné
4. Génération de structures combinatoires, fonction génératrice
   TD : génération uniforme d'arbres
5. Analyse en moyenne d'algorithmes
   TD : analyse de coût moyen, analyse du quicksort
6. Algorithmes randomisés
   TD : calcul de la coupe min

Plan du cours (suite)

Partie II : Espaces continus

1. Génération selon une loi continue
   TD : génération de vecteurs Gaussiens
2. Convergences : loi des grands nombres, fonction caractéristique
   TD : génération de lois continues arbitraires
3. Analyse de données (statistiques descriptives)
   TD : analyse d'échantillon
4. Estimation statistique et intervalles de confiance
   TD : calcul de taille d'échantillon
5. Décision et tests
   TD : test d'adéquation
6. Chaînes de Markov sur un espace discret
   TD : modèles d'automates probabilistes

Ouvrages de référence

Ouvrages de base en probabilité:
Introduction aux Probabilités. P. Brémaud, Springer-Verlag, Berlin, 1984. (version révisée en 97)
Probabilités de l'ingénieur. N. Bouleau, Hermann 1986.
Orienté analyse d'algorithmes :
Analysis of algorithms Sedgewick & Flajolet, Addison Wesley 1996
Orienté évaluation de performances :
The art of computer system performance analysis, Raj Jain, Wiley 1991
Probabilité et Files d'attente :
Probability, stochastic processes, and queueing theory, Randolph Nelson, Springer-Verlag, 1995
Simulation à événements discrets :
Stochastic simulation S.M. Ross, Wiley