

1. Racine carrée entière

La somme des  $n$  premiers impairs est égale au carré de  $n$ .

- (a) Écrire à l'aide de la boucle **tant que** un algorithme permettant de calculer la racine carrée entière d'un nombre entier positif donné.
- (b) Calculer le nombre d'additions et de comparaisons à effectuer.

2. Algorithme du *sac-à-dos*

Une société de logistique appelle un informaticien pour résoudre automatiquement un problème fort simple :

La société doit remplir des conteneurs avec différentes palettes contenant des objets très lourds. Les conteneurs ne doivent pas peser plus de 25 tonnes chacun et chaque palette a un poids et une valeur marchande donnée.

La société doit pouvoir maximiser la valeur marchande de chaque conteneur sans dépasser la charge totale de 25 tonnes. Elle doit donc faire le choix de remplir les conteneurs avec telle ou telle palette sans dépasser le poids total de 25 tonnes et en faisant en sorte que la valeur marchande de chaque conteneur soit la plus grande possible.

- (a) Décrire formellement ce problème
  - Nous nous concentrerons sur un seul conteneur à la fois.
  - Seules la masse et la valeur marchande des objets nous intéressent. Nous nous fichons du volume des objets dans un premier temps par exemple.
  - Le remplissage d'un conteneur peut être assimilé à un vecteur  $C = (x_1, x_2, x_3, \dots, x_k)$  où les objets sont numérotés de 1 à  $k$  et où  $x_i$  vaut respectivement 0 ou 1 si l'objet est écarté ou ajouté au conteneur.
- (b) L'une des solutions pour résoudre ce problème est de considérer tous les vecteurs  $C = (x_1, x_2, x_3, \dots, x_k)$  un à un, et de prendre parmi ceux dont la masse totale est inférieure à 25 tonnes, celui dont la valeur marchande est la plus grande.
  - Combien il y a t-il de vecteurs  $C$  ?
  - Dans quel cas est-ce que cette solution est insatisfaisante ?
- (c) Proposons un autre algorithme :

Les palettes sont triées en fonction de leur valeurs marchandes. On place dans cet ordre, les plus coûteuses d'abord, toutes les palettes qui ne font pas dépasser le poids.

  - Est-ce que cet algorithme donne toujours la bonne solution ?
  - Ecrire en pseudo-code l'algorithme

3. Les tours de Hanoi

Ce jeu consiste à déplacer des disques de diamètres différents d'une tour de départ à une tour d'arrivée en passant par une tour intermédiaire.

Les deux règles à respecter sont les suivantes :

- On ne peut déplacer plus d'un disque à la fois.
  - On ne peut pas placer un disque sur un autre disque plus petit.
- (a) Décrire le problème à résoudre pour  $n$  disques en faisant l'hypothèse qu'il a déjà été résolu pour  $n - 1$  disques et qu'il est trivial pour un seul disque.
  - (b) Écrire un algorithme permettant de résoudre récursivement le problème pour un nombre quelconque de disques.
  - (c) Calculer le nombre de déplacements à effectuer pour un nombre de disques donné.

4. Coefficient binomial d'entiers

Nous rappelons que le coefficient binomial des entiers  $n, p$  noté  $C_n^p$  vaut  $\frac{n(n-1)(n-2)\dots(n-p+1)}{p!}$  est donné par la formule de Pascal :

$$C_n^p = C_{n-1}^{p-1} + C_{n-1}^p$$

Nous rappelons par ailleurs que  $(x + y)^k = \sum_{k=0}^n C_n^k x^{n-k} y^k$

- (a) Écrire un algorithme récursif qui permet de calculer le coefficient binomial de deux entiers donnés.
- (b) Quelle est le nombre d'additions pour calculer  $C_n^p$  avec cette méthode ?
- (c) Écrire un algorithme qui permet d'écrire le développement de  $(x + y)^k$