# Corso di Laurea in Informatica
# Corso di Algoritmi
### Esercizi proposti – 9 Marzo 2018

**Per ognuno dei seguenti problemi:**

- **a)** Scrivere la definizione induttiva.
- **b)** Fornire l'algoritmo di programmazione dinamica basato sulla definizione data.
- **c)** Specificare la complessità in tempo dell'algoritmo a costi uniformi nel caso peggiore.

## 1. Torneo

In un torneo si sfidano due squadre: INF e MAT. I due team giocano al più 2n-1 partite, in quanto vince la squadra che per prima raggiunge n vittorie, supponendo che non vi siano partite che finiscono alla pari.
Si conosce la probabilità p che ha la squadra INF di vincere in ogni singola partita, mentre q = (1- p) è la probabilità di vincere in ogni singola partita della squadra MAT.
Indichiamo con P(i, j) la probabilità del team INF di vincere il torneo quando gli mancano i vittorie, mentre all'avversario ne mancano j per aggiudicarsi il torneo. P(n, n) è pertanto la probabilità iniziale, che ci interessa conoscere.
Nel caso i, j > 0 la probabilità P(i, j) può essere espressa in funzione di P(i-1, j) e P(i, j-1) nel modo seguente: $P(i, j) = p \cdot P(i-1, j) + q \cdot P(i, j-1)$

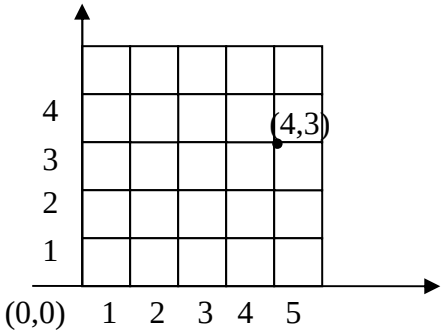
## 2. Resto

Si hanno monete di n valori diversi e T[1..n] è un array che fornisce tali valori; per ogni valore si dispone di un numero illimitato di monete. Si deve scegliere il minimo numero di monete il cui valore complessivo ammonti ad un intero dato, minore o uguale ad L.

N.B. Si usi ∞ per indicare l'impossibilità di formare la somma con le monete a disposizione.

## 3. Percorso a Manhattan

A Manhattan le strade si incrociano ad angolo retto e i block (gli isolati) hanno le stesse dimensioni, che indicheremo con 1x1. Vogliamo sapere quanti sono i percorsi di lunghezza minima tra due incroci.
Mettendo la griglia di strade su un piano cartesiano possiamo esprimere il problema nel modo seguente: dato un punto P(m, n) di coordinate intere, determinare il numero di spezzate di lunghezza minima che congiungono P al punto (0, 0).



## 4. Catene di montaggio .

Uno stabilimento automobilistico ha due catene di montaggio. Il telaio di un'automobile entra in una catena, riceve nuovi componenti in un certo numero di stazioni e infine esce completa dalla catena di montaggio. Ogni catena ha n stazioni: $S_{i,j}$ (i = 1, 2, j = 1, …, n). Le stazioni j-esime su ognuna delle due catene eseguono le stesse lavorazioni, ma essendo state costruite con tecnologie diverse, i tempi di lavorazione sono diversi. Sia $a_{i,j}$ il tempo di montaggio nella stazione $S_{i,j}$. Il telaio impiega un tempo $x_i$ per entrare nella prima stazione della catena i, i tempi di spostamento da una stazione alla successiva della stessa catena sono trascurabili, mentre si ha un tempo di uscita $y_i$ per l'automobile completa.
Quando si presenta un ordine urgente la macchina in lavorazione può essere trasferita da una catena ad un'altra per minimizzare i tempi; in tal caso comunque si paga un tempo di trasferimento $t_{i,j}$ per abbandonare la catena i, in uscita dalla j-esima stazione.
Il problema consiste nel determinare quali stazioni di lavorazione scegliere dalla catena 1 e quali dalla catena 2 in modo da minimizzare il tempo totale di assemblaggio dell'automobile.

## 5. Concatenazione di stringhe

Dato un insieme di stringhe binarie $\{Y_1, Y_2, \ldots, Y_m\}$, dette *primitive*, ed una stringa binaria X, si vuole determinare se la stringa X si può ottenere dalla concatenazione di stringhe primitive. Ad esempio: dato l'insieme di primitive {01, 10, 011, 101}, per la stringa X = 0111010101 la risposta è si (ci sono 2 possibili soluzioni) mentre per la stringa X = 0110001 la risposta è no. Si supponga di avere per ogni primitiva $Y_i$ l'attributo length($Y_i$), che ne fornisce la lunghezza, e di disporre di una funzione *test* ($x_r$, $x_{r+1}$, …, $x_s$, y), che restituisce *vero* se i caratteri $x_r\, x_{r+1} \ldots x_s$ sono uguali alla stringa y, *falso* altrimenti.