

# Esercizi4

## Es. 1,2, 3, 4

giocando  $m$  partite con 3 carte ci aspettiamo di vincere  $(1/3 \cdot 2 + 1/3 \cdot 2)m$  soldi spendendo  $1 \cdot 1/3$

Giocando  $m$  partite con  $n$  carte e vincendo  $v$ €, se si decide di girare al massimo  $k$  carte si avrà:

Spesa attesa quando si perde

$$\frac{n-k}{n} k$$

$n-k$  volte su  $n$  non vinco e quindi spendo  $k$  soldi

vincita attesa:

$$\frac{k}{n} v = \frac{kv}{n} \text{ euro}$$

Spesa attesa quando si vince

(Si vince alla  $k$ -esima carta):

$$\sum_{i=1}^k \frac{1}{n} i = \frac{1}{n} \sum_{i=1}^k i = \frac{1}{n} \frac{k(k+1)}{2} = \frac{k(k+1)}{2n}$$

euro

Guadagno:

$$\frac{kv}{n} - \frac{k(k+1)}{2n} - \frac{k(n-k)}{n} = \frac{2kv - k(k+1) - 2k(n-k)}{2n} = \frac{k(k}{2n}$$

Il valore intero di  $1 \leq k \leq n$  che massimizza  $\frac{k(k+2v-2n-1)}{2n}$  è la migliore strategia

## Es. 5

$$F(n) = 2S\left(\frac{n}{2}\right) + c * n$$

$$S(n) = F(n - 1) + c * n$$

Sequenza:

5, 1,

L'insieme di elementi nella sequenza sono i numeri naturali positivi compresi largamente tra 1 e 9.

Se al primo round il primo pivot è proprio 5, allora i due sotto-problemi sono di taglia 4 rispettando il vincolo

$$F(n) = 2S\left(\frac{n}{2}\right) + c * n.$$

Se per il secondo caso vengono scelti rispettivamente 1 e 9 abbiamo che i due sotto problemi saranno di taglia 3 rispettando il vincolo  $S(n) = F(n - 1) + c * n$ .

Nell'ultima fase si applica la forza bruta in quanto ci sono solo 3 elementi.

## Es. 6

Analogamente a *RandomQuickSortBalanced*, se vogliamo bilanciare il *RandomSelect* dobbiamo iterare la scelta del

valore  $i$  fin quando uno tra  $S^<$  o  $S^>$  avrà cardinalità almeno  $\frac{n}{k}$  con  $k$  arbitrario.

$$\min(|S^<|, |S^>|) \geq \frac{n}{k}$$

$$E(T) = E(T(\frac{n}{k})) + 2cn$$

oppure

$$E(T) = E(T(\frac{(k-1)n}{k})) + 2cn$$

(Se prendiamo la parte peggio sbilanciata)

$2cn$  dipende da quante volte vogliamo iterare la ricerca del valore  $i$ . Bisogna tenere in considerazione che ci sono  $\frac{k-2}{k}$  valori centrali.

Siccome ci troviamo sempre in una situazione in cui dobbiamo analizzare una frazione dell'input, la complessità di tempo è  $O(\log(n))$

## Es. 7

Sia  $G$  un grafo dove  $E$  è l'insieme dei lavori  $w$  e  $V$  è l'insieme dei lavoratori  $d(w_i)$ .

Assegniamo le competenze in modo casuale.

$d_1$  e  $d_2$  avranno un arco se e solo se le loro competenze sono diverse.

Possiamo applicare un algoritmo per trovare il taglio minimo (o massimo flusso).

Il taglio minimo ci darà il numero massimo di lavori eseguibili.

Questo ci dirà pure se abbiamo assegnato in modo buono le competenze a ciascun nodo del grafo.

Sia  $X_i$  così definita:

$$X_i = 0 \text{ se } d_1(w_i) = d_2(w_i)$$

$$X_i = 1 \text{ altrimenti.}$$

$$Pr[X_i = 0] = \frac{1}{9} * 3 = \frac{1}{3}$$

$$Pr[X_i = 1] = 1 - \frac{1}{3} = \frac{2}{3}$$

$$E[X] = E[X_1] + \dots + E[X_n] = (0 * \frac{1}{3} + 1 * \frac{2}{3})n \geq \frac{2}{3}n$$

## Es. 8

Se ci sono  $n$  letterali in una clausola, la probabilità che la clausola  $C_i$  sia falsa è  $\frac{1}{2^{k_i}}$ .

Ovviamente dobbiamo calcolare il valore atteso.

$n$  è il numero di clausole.

$$Z_i = 1 \text{ se } C_i \text{ è soddisfatta}$$

$$Z_i = 0 \text{ altrimenti}$$

$$E[Z] = E[Z_1] + \dots + E[Z_n]$$

$$E[Z_i] = (1 - \frac{1}{2^{k_i}})$$

$$E[Z] = (1 - \frac{1}{2^{k_1}}) + \dots + (1 - \frac{1}{2^{k_n}}) \geq (1 - \frac{1}{2^{k_j}})n$$

Dove  $C_j$  è la clausola con il minor numero di letterali.

La probabilità di ottenere la soluzione migliore dipende quindi dal minor numero di letterali nelle clausole.

## Es. 9

No perché non potremmo esplorare tutto l'insieme dei possibili super-nodi  $s$  e  $t$ .