****

DIPARTIMENTO DI SCIENZE MATEMATICHE, INFORMATICHE E FISICHE

**PERIODO FRAZIONARIO**

**Implementazioni e analisi delle complessità**

Cappa Davide [151953@spes.uniud.it](mailto:151953@spes.uniud.it) 151953

Marinato Federico [157893@spes.uniud.it](mailto:157893@spes.uniud.it) 157893

Raffin Matteo [156911@spes.uniud.it](mailto:156911@spes.uniud.it) 156911

Obbiettivo: Implementazione e analisi della complessità sul calcolo del periodo frazionario minimo tra stringhe, sviluppando l’algoritmo in due versioni differenti.

Il periodo frazionario di una stringa s è il più piccolo intero p>0 che soddisfa la proprietà seguente: ∀i = 1,…,n−p s(i)=s(i+p)

Le due versioni dell’algoritmo sono:

* *PeriodNaive*
* *PeriodSmart*

*PeriodNaive*, partendo da q=1, ovvero il minimo periodo ammissibile, fino ad al massimo q=n, ovvero la lunghezza della stringa, verifica se esso è effettivamente il periodo della stringa, scandendola in modo diretto; quando ciò accade l’algoritmo termina restituendo il valore di q finale.

*PeriodSmart* risolve il problema sfruttando il concetto di bordo calcolato progressivamente per induzione ad ogni variazione di n per ottenere una soluzione efficiente.

La lunghezza n è compresa tra un range di valori fra 100 e 50000 con distribuzione esponenziale. Tale n dunque è definita dalla seguente funzione: n(i):=floor( A\*(B^i) ) dove i varia nell’insieme {0,…,99}, mentre A e B sono tali che n(0)=100 e n(99)=50000.

Una stringa è generata tramite l’alfabeto ternario {‘a’,’b’,’c’} nel seguente modo: nel caso medio i primi q caratteri, dove q rappresenta il periodo, sono generati in modo pseudo-casuale e indipendentemente tra loro mentre la parte rimanente è costruita seguendo la formula *s*(*i*)=*s(i-q)* per ogni *i*=*q*+1,…,*n*. Per ogni lunghezza n della stringa s, q avrà un valore tra 1 ed n, ottenuto in maniera aleatoria, prima di generare la sequenza s. Per costruire la stringa relativa al caso pessimo nella versione *PeriodNaive* è stato utilizzato un alfabeto binario {‘a’,’b’}, in modo che l’unico carattere diverso dagli altri fosse quello in ultima posizione.

La macchina su cui si sono effettuate le misurazioni ha le seguenti specifiche: …

Per l’analisi dei tempi occorre prima di tutto calcolare la risoluzione R del clock di sistema, ovvero l’intervallo minimo di tempo misurabile. Nel nostro caso la risoluzione è pari a …. . Per tale valore di R il tempo minimo misurabile, ovvero il tempo di esecuzione affinché l’errore relativo generato è minore di quello voluto, è calcolato come Tmin=R\*(1/E + 1) dove E è l’errore relativo massimo ammissibile (0.001). Per tutte le misurazioni del tempo trascorso è necessario utilizzare un clock di sistema monotono, il quale sarà richiamato 2 volte: rispettivamente prima e dopo l’esecuzione del calcolo del periodo frazionario con *PeriodNaive* e/o *PeriodSmart*.

Per ottenere delle misurazioni accurate nel caso medio, è stata considerata, oltre al tempo medio, anche la mediana e la deviazione standard in funzione della lunghezza della stringa n. Per ogni lunghezza n, si è andati a generare 20 sequenze diverse come campione di dati per il calcolo degli indici espressi precedentemente (ciò non si rende necessario per il caso pessimo). Per il calcolo della mediana è stato ordinato il vettore contenente i tempi delle misurazioni e si è individuato il valore mediano. Mentre la deviazione standard è stata calcolata rispetto al valore medio. I tempi sono stati misurati in ms con un numero di 7 cifre significative.

Per la rappresentazione grafica dei tempi di *PeriodNaive* e *PeriodSmart* si sono utilizzate due scale differenti, una lineare e una logaritmica: con la prima si vede l’andamento effettivo, mentre la seconda, essendo le lunghezze n generate con distribuzione esponenziale, evidenzia meglio il comportamento asintotico. Si osserva nei seguenti casi:

Caso medio:

In entrambi i grafici relativi al caso medio, al crescere della lunghezza n della stringa il tempo non cresce in maniera direttamente proporzionale. Il tratteggio rappresenta la deviazione standard, calcolata rispetto il tempo medio.

Si nota che entrambi gli algoritmi hanno un andamento abbastanza lineare: *PeriodNaive* risulta essere leggermente più efficiente di *PeriodSmart,* pur mantenendo la stessa complessità asintotica; si potrebbe dunque ipotizzare che in generale le costanti moltiplicative del primo siano più piccole. Sono visibili in particolare dei casi in cui per certe lunghezze n delle stringhe, la differenza dei tempi di esecuzione è ingente; si può supporre che in queste situazioni il periodo delle stringhe generate è molto grande per cui in *PeriodNaive* il ciclo annidato viene svolto meno frequentemente e in *PeriodSmart …. (condizione di arresto periodSmart).* Analizzando la deviazione standard non c’è una simmetria tra lo scarto positivo e negativo al valore mediano, conferma del fatto che è stata calcolata rispetto al valore medio. Non c’è correlazione tra lo scarto e il crescere della lunghezza della stringa.

Entrambi gli andamenti sono abbastanza lineari, infatti non si rilevano particolari differenze tra *PeriodNaive* e *PeriodSmart,* per la verifica della proprietà enunciata prima (…) si è utilizzato un ciclo secondario controllando in modo diretto che s(i)=s(i+p) per ogni i=1,…,n−p. Rispetto alla scala lineare non si notano i picchi fra i tempi medi. Al crescere della lunghezza n la deviazione standard diminuisce, indice …

Caso pessimo:

Per i grafici a seguire, che rappresentano i casi pessimi non è stata evidenziata la deviazione standard in quanto per ogni lunghezza n la costruzione della stringa non è casuale, ma segue la descrizione data. Quindi si ha una sola stringa per ogni n e dunque una sola misurazione.

*PeriodNaive* risulta avere un andamento fortemente quadratico, e questo comportamento è messo in evidenza dalla scala usata nel grafico sopra che però nasconde il comportamento di *PeriodSmart*, che si sa essere lineare anche al caso pessimo.

Il grafico risulta essere molto diverso da quello in scala logaritmica del caso medio: ora per entrambi gli algoritmi si verificano andamenti approssimabili a due rette. Quest’ultime, sono di coefficiente angolare differente, in particolare quello della retta di *PeriodNaive* è circa il doppio ( da dimostrare…). *PeriodSmart*, come già detto, è più efficiente dal punto di vista del tempo, tuttavia, la sua retta ha un andamento più a “zig zag” mentre la retta di *PeriodNaive* dopo un certo n ha valori che seguono più fedelmente una retta.

Conclusione: Si è calcolato il periodo frazionario tra 2 stringhe utilizzando *PeriodNaive* e *PeriodSmart.* Dai grafici è emerso che *PeriodSmart* in ogni casistica è lineare…, tuttavia in esecuzione utilizza dello spazio aggiuntivo dell’ordine di n. Mentre *PeriodNaive* èquadratico nel caso peggiore, lineare nel caso medio ma non utilizza spazio aggiuntivo.

TODO:

* Esplicitare 100 lunghezze generate
* Prestazioni Pc
* Spiegazioni grafici(scala lineare, scala logaritmica, differenze casi pessimo e medio e perché)
* Casi particolari di grafici
* Conclusioni