**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 2 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 2 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 naive è stato utilizzato un alfabeto binario {‘a’,’b’}, in modo che l’unico carattere diverso dagli altri fosse quello in ultima posizione.

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, 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. 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 ms con un numero di … cifre significative.

Caso medio (scala lineare):

Caso medio (scala logaritmica):

Caso pessimo (scala lineare):

Caso pessimo (scala logaritmica):

TODO:

* Calcolo dei tempi
* Scrivere del calcolo della mediana del valore:
* Scarto quadratico medio
* Spiegazioni grafici(scala lineare, scala logaritmica, differenze casi pessimo e medio e perché)
* Casi particolari di grafici
* Conclusioni