

Progetto Laboratorio Algoritmi e Strutture Dati (Periodo Frazionario)

Lorenzo Bussi 149462@spes.uniud.it
Fabio Ionut Ion 157274@spes.uniud.it

Luglio 2023

Indice

1	Introduzione	1
2	Analisi caso migliore	2
3	Analisi caso medio	3
4	Analisi caso peggiore	4
5	<i>PeriodNaive</i>: caso migliore, medio e pessimo a confronto	6
6	<i>PeriodSmart</i>: caso migliore, medio e pessimo a confronto	7
7	Conclusioni	8

1 Introduzione

Per risolvere il problema del calcolo del periodo frazionario minimo di una stringa ci è stato richiesto di implementare due algoritmi: *PeriodNaive* e *PeriodSmart*, rispettivamente con complessità $\Theta(n^2)$ e $\Theta(n)$.

Durante l'implementazione dell'algoritmo *PeriodNaive* abbiamo deciso di utilizzare il primo metodo descritto sulla pagina E-Learning, motivo per cui questo potrebbe avere tempi lineari nel caso medio, invece che quadratici.

Per quanto riguarda l'implementazione dei due algoritmi in esame e l'algoritmo per la misurazione dei tempi d'esecuzione di questi è stato usato il linguaggio C. La realizzazione dei grafici che mettono a confronto i vari casi di esecuzione di entrambi gli algoritmi è stata invece effettuata avvalendoci del software open source *gnuplot*. Al fine di visualizzare con più precisione i tempi di esecuzione nei vari casi abbiamo impiegato per ogni grafico sia la scala lineare che logaritmica.

2 Analisi caso migliore

Il caso migliore si verifica quando i due algoritmi operano su stringhe nella configurazione "aa...aa", ossia stringhe costruite su un alfabeto con solamente il carattere 'a'.

Sono state generate quindi 100 stringhe, tutte di lunghezza diversa nell'intervallo [1000, 500000] - usando come alfabeto {'a'} - e sono stati misurati i diversi tempi di esecuzione per entrambi gli algoritmi.

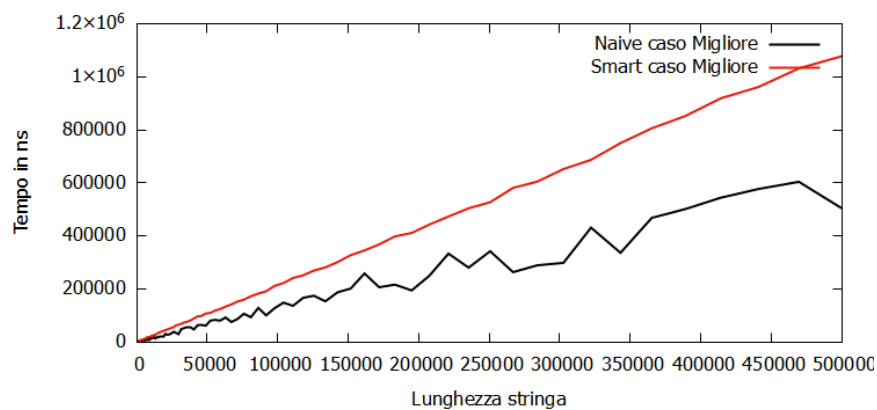


Figura 1: scala lineare caso migliore

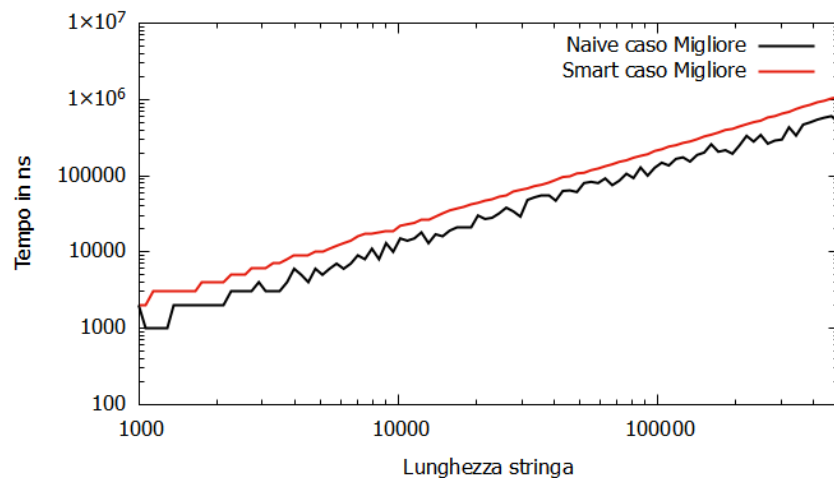


Figura 2: scala doppiamente logaritmica caso migliore

Commenti I grafici Figure 1 e Figure 2 mettono a confronto l'esecuzione di *PeriodNaive* e *PeriodSmart* prendendo in input lo stesso set di stringhe nel formato "aa...aa".

Si può notare come in questo caso l'algoritmo *PeriodNaive* sia leggermente più efficiente rispetto a *PeriodSmart*, sebbene entrambi presentino comunque un andamento lineare.

3 Analisi caso medio

I due algoritmi sono stati testati prendendo in input stringhe costruite in modo pseudo-casuale su un alfabeto di tre caratteri {'a','b','c'}.

La generazione delle stringhe in questo caso è stata più complessa rispetto al caso precedente, in quanto abbiamo deciso di generare 20 campioni diversi di stringhe per ogni possibile lunghezza. Abbiamo quindi cercato di rendere l'analisi più accurata, andando a scegliere il mediano tra le 20 misurazioni effettuate per ogni diversa dimensione di stringa.

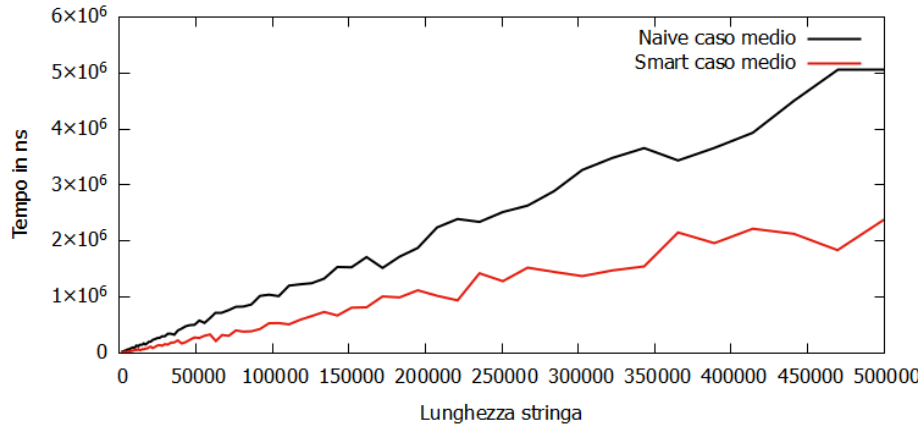


Figura 3: scala lineare caso medio

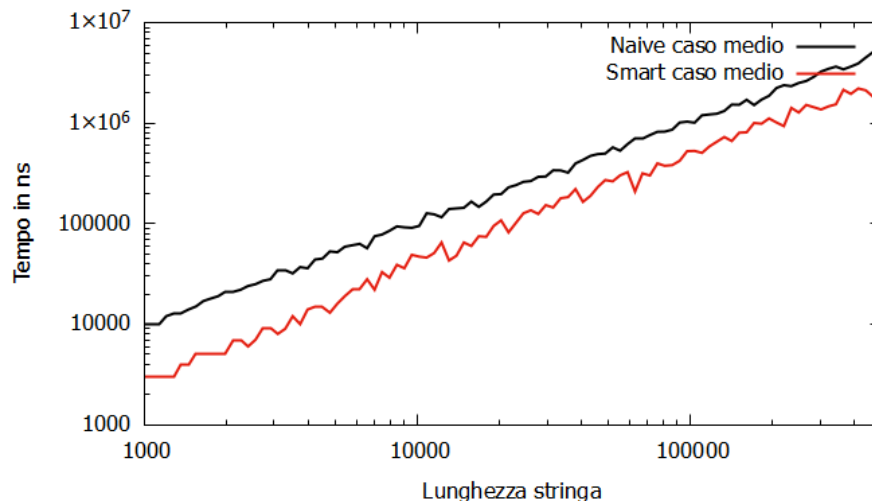


Figura 4: scala doppiamente logaritmica caso medio

Commenti I grafici Figure 3 e Figure 4 mettono a confronto l'esecuzione di *PeriodNaive* e *PeriodSmart* prendendo in input lo stesso set di stringhe. In questo caso notiamo che l'algoritmo *PeriodSmart* è leggermente più efficiente rispetto a *PeriodNaive*, seppure entrambi presentino un andamento lineare. In linea con quanto prospettato, l'algoritmo *PeriodNaive* mostra un andamento lineare nel caso medio e non quadratico, avendo implementato l'algoritmo come specificato nell'Introduzione.

4 Analisi caso peggiore

Il caso peggiore si verifica quando le stringhe sono costruite su un alfabeto di due soli caratteri $\{'a', 'b'\}$ nella configurazione "aa...aab". Osserviamo che le stringhe con questa configurazione non portano al caso peggiore per quanto riguarda *PeriodSmart*, dal momento che le stringhe dovrebbero trovarsi in una configurazione tale da non generare nessun periodo. Abbiamo scelto, per semplicità, di considerare solamente il caso in cui le stringhe dispongono della seguente forma "aa...aab".

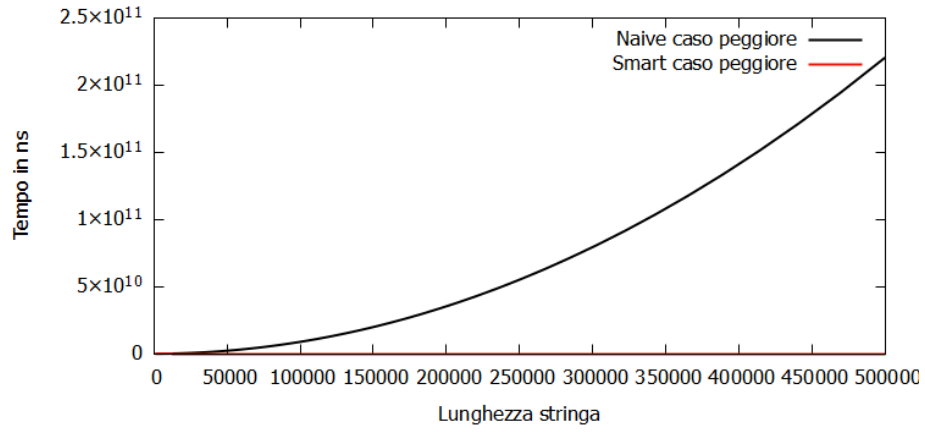


Figura 5: scala lineare caso peggiore

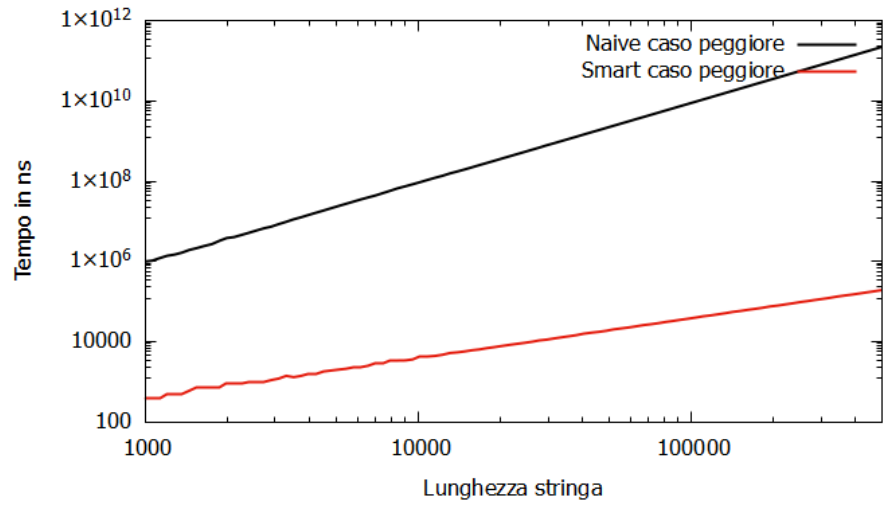


Figura 6: scala doppiamente logaritmica caso peggiore

Commenti Dai grafici si può concludere che l'algoritmo *PeriodNaive* presenta un andamento quadratico rispetto alla lunghezza n della stringa, in linea con quanto pronosticato; mentre *PeriodSmart* ha andamento apparentemente costante, ma di fatto lineare.

5 *PeriodNaive*: caso migliore, medio e peggior a confronto

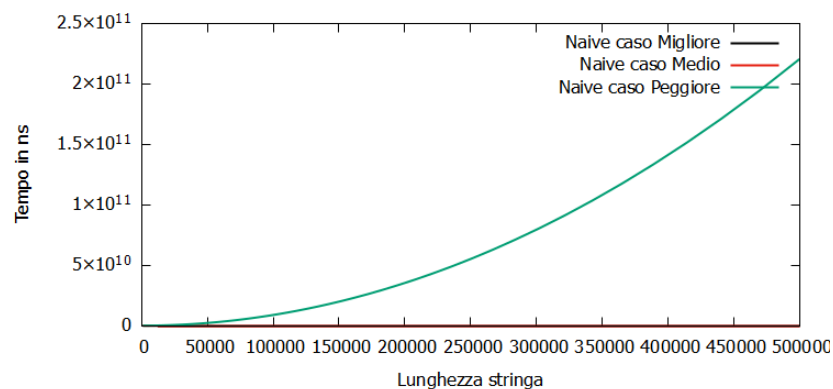


Figura 7: scala lineare

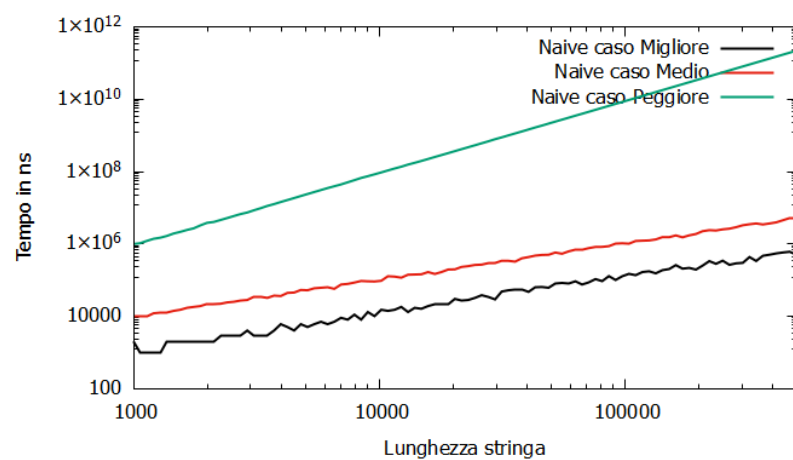


Figura 8: scala doppiamente logaritmica

6 *PeriodSmart*: caso migliore, medio e peggior a confronto

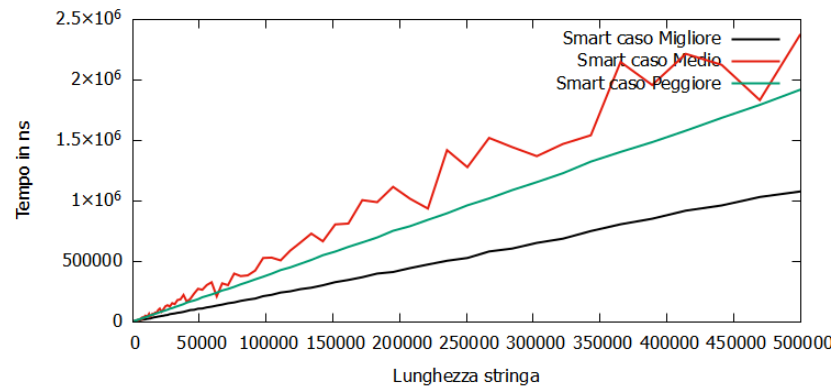


Figura 9: scala lineare

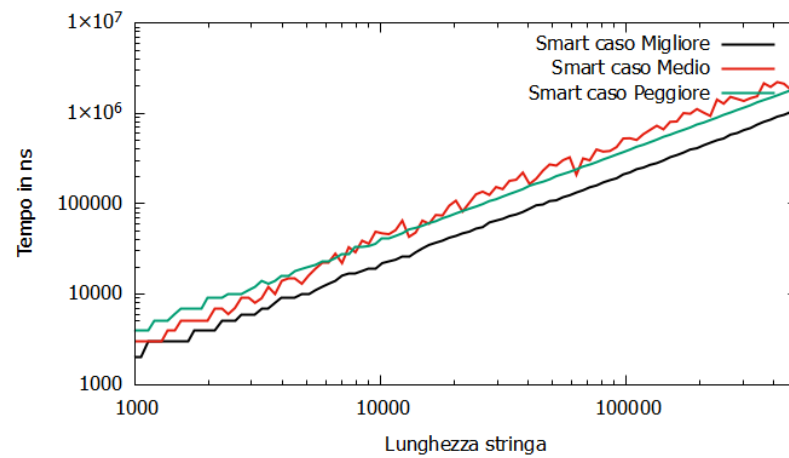


Figura 10: scala doppiamente logaritmica

7 Conclusioni

I risultati ottenuti ci hanno permesso di concludere che

1. *PeriodSmart* non ha un vero e proprio caso peggiore e migliore, in quanto i dati confermano che la complessità è $\Theta(n)$, visto che la funzione di crescita ha un andamento lineare qualsiasi sia l'input.
2. L'algoritmo *PeriodNaive* ha complessità quadratica solo nel caso peggiore: dovuto al modo in cui abbiamo deciso di implementarlo, altrimenti avremmo osservato complessità $\Theta(n^2)$ già nel caso medio.
3. Entrambi gli algoritmi presentano almeno un caso con complessità lineare