

# SNEIC



Series of **N**umbers by **E**xploration and **I**nhibition **C**onstructor

by Stefano de Martino

SNEIC prende in input un insieme di combinazioni della stessa lunghezza (ad esempio un sistema ridotto oppure un archivio di concorsi) e serve per la ricerca di combinazioni (lungchette) che in NON abbiano in comune, con nessuna combinazione di questo insieme, un certo numero di punti.

Può quindi essere utilizzato, ad esempio, dato un sistema ridotto che vogliamo verificare come input, per verificare se in esso esistono una o più combinazioni che NON totalizzano 2 punti. Oppure, dato un archivio di estrazioni di una lotteria, per cercare tutte le lungchette (di dimensione specificata dall'utente, ad esempio di 15 numeri) che non hanno mai totalizzato 4 o più punti.

**1** **Seleziona il file** con le estrazioni (oppure il sistema) che vuoi esaminare

**2** **Imposta i parametri della tua ricerca** (quanti numeri vuoi cercare e quanti punti questi numeri NON devono totalizzare nel tuo file)

**3** **Scegli la cartella e il nome del file per i risultati** della ricerca

**4** Infine **AVVIA** la ricerca

The screenshot shows the SNEIC application window with the following settings: File di input (concorsi o sistema) is selected; Lunghezza cercata: 9; Minimo punti rifiutati: 3; Numero max da provare nelle lungchette (max in integrale): 90; Imposta Intervallo di ricerca is checked; primo numero sequenze, Da: 1 a 90; File di Output (lungchette trovate) is selected; Salva anche i tempi di ciascuna lungchetta trovata is unchecked; Buttons: Avvia ricerca (green), Stop (red), ABILITA SUONI (checked), Test volume; Table with columns: Nr. combinazioni file, Formazione, Massa n., Min, Max; Status: fase; Timing: Ora Inizio h.m.s, Ora Fine h.m.s, Trascorso; Results: Lunghezza max trovata finora: -- (#), Tentativi effettuati: --, Trovate: 0.

L'interfaccia utente è progettata per fornire tutte le informazioni necessarie sulla ricerca che si sta effettuando.

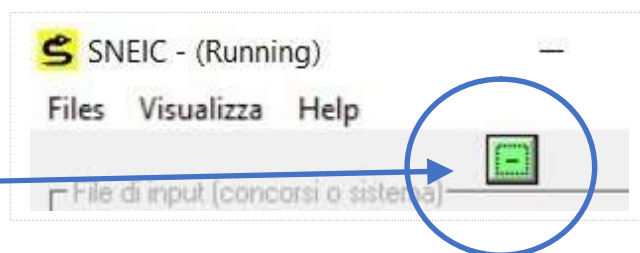
È possibile **visualizzare in tempo reale i risultati intermedi** (anche se la ricerca non è ancora in corso)



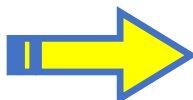
È possibile **attivare gli avvisi sonori:** uno quando viene trovata una soluzione (lunghetta), l'altro quando tutte le possibilità sono state esplorate e quindi l'elaborazione termina.



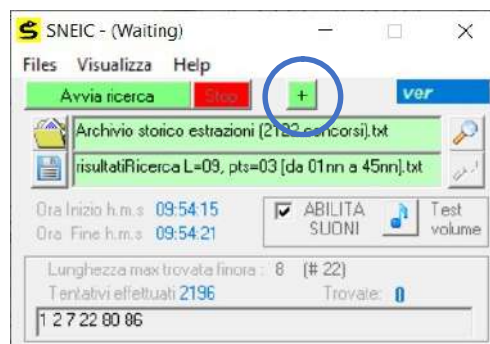
Si ha la possibilità di **ridimensionare la finestra** in modo da ridurre lo spazio occupato sullo schermo durante l'elaborazione mantenendo il controllo sullo stato di avanzamento.



Finestra con form STANDARD

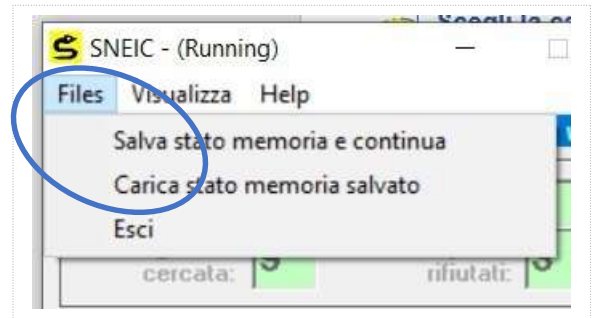


Finestra con form PICCOLO



Si possono **sospendere e salvare le ricerche in corso** in una cartella in modo da poter **proseguire successivamente** la stessa ricerca esattamente dal punto in cui era stata salvata.

Tramite questa funzione è anche possibile proseguire il lavoro su un altro computer.



È possibile restringere l'**intervallo d ricerca** specificando dove la ricerca deve iniziare e dove terminerà. Per farlo si deve inserire il primo numero della lunghetta iniziale e di quella finale (non è possibile specificare la lunghetta per intero, ma solo la **prima cifra** della **prima lunghetta** e la **prima cifra** per l'**ultima lunghetta**).

Ad esempio:

Sequenza PARTENZA **1** 2 3 4 5 6 7 8 9,      sequenza ARRIVO **1** 83 84 85 86 87 88 89 90



oppure:

Sequenza PARTENZA **2** 3 4 5 6 7 8 9 10,      sequenza ARRIVO **10** 83 84 85 86 87 88 89 90



Quest'ultima opzione rende **possibile spezzare una ricerca in vari processi separati** dello stesso programma sullo stesso file di Input (applicando, di fatto, l'implementazione manualmente di vari threads contemporaneamente sullo stesso problema).

Quindi, invece di lanciare una lunga ricerca che analizzi le possibili lunghette dalla prima all'ultima (*ad esempio, per una ricerca di nove elementi la prima lunghetta sarebbe 1 2 3 4 5 6 7 8 9 e l'ultima 82 83 84 85 86 87 88 89 90*) si potrebbe scegliere di dividere questo intervallo in due, tre, otto parti e anche di più.

Come negli **esempi qui sotto**, in cui non è stato deciso un criterio preciso di scomposizione del problema (la suddivisione è stata fatta a caso):

**Quattro istanze** contemporanee che lavorano sullo stesso file, ma su intervalli di ricerca sull'integrale differenti, definiti dalla prima cifra delle lunghette di inizio e fine:

- 1° da 1 a 1 (ovvero dalla lunghetta 1 2 3 4 5 6 7 8 9 alla lunghetta 1 83 84 85 86 87 88 89 90)
- 2° da 2 a 2 (ovvero dalla lunghetta 2 3 4 5 6 7 8 9 10 alla lunghetta 2 83 84 85 86 87 88 89 90)
- 3° da 3 a 3 (ovvero dalla lunghetta 3 4 5 6 7 8 9 10 11 alla lunghetta 3 83 84 85 86 87 88 89 90)
- 4° da 4 a 4 (ovvero dalla lunghetta 4 5 6 7 8 9 10 11 12 alla lunghetta 4 83 84 85 86 87 88 89 90)



per aggiungere molte finestre sullo schermo è utile ridurre le dimensioni di ciascuna istanza per poter seguire l'avanzamento di ciascuna di esse:



Il numero dei processi in esecuzione contemporaneamente può aumentare in rapporto alla velocità del computer in uso e del numero di processori (Core e Threads della CPU). Ma questo algoritmo ha un uso molto ridotto della memoria ram perciò, potenzialmente si possono avviare davvero molte istanze contemporaneamente anche su computer più vecchi.