Risoluzione automatica del cruciverba

Progetto di Esperienze di programmazione

Davide Coffaro Matricola 556603

2020

Il problema di “Risoluzione automatica del cruciverba” è quello di riempire lo schema dato (con caselle bianche e caselle nere) con le parole disponibili, partendo da un’unica parola inserita nello schema.  
Questo problema può essere visto anche come problema che data una parola con l’iniziale o solo alcune lettere, proponga in automatico le parole disponibili (ricercandolo all’interno di una lista di parole), dando come input anche la lunghezza della parola voluta oppure con una procedura diversa tutte le parole disponibili non tenendo conto della lunghezza.

Algoritmi per risolvere il problema

* Algoritmo1 - parto dallo schema, cerco la parola da inserire fino a completare il cruciverba:

INPUT schema iniziale + lista parole da inserire  
OUTPUT cruciverba completato o no (true o false)

PROCEDURA ALGORITMO1:

1. Inizializzo trovato(i)=false per i=2…n, questo indicherà per le caselle di lunghezza i se sono state trovate tutte le parole. Inizializzo il timer tempoDiEsecuzione;
2. Finche trovato(i)=true per ogni i=2…n, tempoDiEsecuzione<tempoMax, cicliEseguiti<cicliMax;
3. Incremento i cicliEseguiti di 1. Prendo le parole dello schema di c numero di caselle e le inserisco in una lista (controllando solo le parole ancora da completare), con c=2…n, se c=n+1 torno al punto 2);
4. Fino a che la lista di parole trovate di c caselle non è vuota, se è vuota torno al punto 3) ricercando parole con numero di caselle c+1;
5. A parità di c caselle prendo la parola x con più lettere già inserite;
6. Controllo quali parole di c lettere possono essere inserite nello spazio della parola x con c caselle, facendo il confronto delle substringhe lettere-caselle;
7. Se è stata trovata una parola la inserisco nello schema e torno al punto 4), altrimenti torno al punto 4), prendendo la prossima parola di c caselle se presente.

CONDIZIONI DI TERMINAZIONE:  
1) trovato(i)=true per i=2…n;  
2) tempo di esecuzione algoritmo>tempoMax???;  
3) cicliEseguiti>cicliMax.

* Algoritmo2 - Parto invece dalle parole disponibili nella lista delle parole da inserire, le analizzo e le inserisco fino a completare il cruciverba:

1)

* Algoritmo4 – procedura simile all’algoritmo1 ma:

1. Utilizzo di lista(i) che contengono le parole di lunghezza i per i=1…n
2. Quando inserisco la parola di lunghezza i nello schema, la elimino anche dalla sua lista(i)
3. Quando lista(i) è vuota per i=1…n termino il ciclo

Problemi relativi all’algoritmo4:

* + Riempire lista(i) all’inizio durante la creazione delle caselle ecc…
  + Memoria occupata da listaParolaSchema, listaParoleSchemaDaCompletare, listaParoleDaInserire (dizionario input)

Risultati ottenuti:

* Credo che valga per tutti gli algoritmi, il dizionario delle parole disponibili/da inserire alla fine dell’elaborazione non è vuoto ma risulta contenere alcune parole le cui lettere sono state tutte inserite durante l’inserimento di altre parole, quindi la parola è completa ma non è stata inserita nello schema tramite la procedura degli algoritmi. Questo fa sì che ci sia una procedura di aggiornamento del dizionario delle parole disponibili che cicla tutte le parole inserite nello schema che sono complete e le elimini dal dizionario se sono contenute al suo interno. AUMENTA LA COMPLESSITà dell’algoritmo/algoritmi.