

# Super Sopa

GRAU A      Q1 CURS 2022-2023

Departament de Ciències de la Computació  
Universitat Politècnica de Catalunya

## Resum

Aquest projecte té com a objectiu realitzar una validació experimental de l'eficiència i aplicabilitat de diferents algorismes i estructures de dades per a cercar paraules a una sopa de lletres estrambòlica que anomenarem *Super Sopa*.

El projecte es farà en grups de **4 persones**, formats (en la mesura del possible) per estudiants matriculats al mateix subgrup de problemes. Per formalitzar els grups us heu d'apuntar al fitxer compartit **GRAU-A - Grups de Projecte** (seguiu l'enllaç).

El lliurament dels materials demanats a aquest projecte es faran en línia via el **Racó FIB** i s'haurà de dur a terme abans de les 23:59 hores del dia **16 d'octubre de 2022**.

En qualsevol moment durant el procés de correcció podríeu ser contactats per part d'algun professor de l'assignatura per tal de resoldre dubtes o fer aclariments sobre el vostre treball.

Totes les comunicacions públiques referents al projecte es duran a terme mitjançant el **canal d'Slack #projecte**.

## 1 El Joc

Disposem d'una col·lecció de paraules a la qual denominarem *Diccionari*. Les *paraules* del diccionari són cadenes de caràcters alfabètics de qualsevol longitud.

Per altra banda tenim una Super Sopa, que és un *tauler* de  $n \times n$  caselles. A cada casella s'hi troba una lletra de l'alfabet. Dues caselles del tauler són *adjacents* si són contigües horitzontalment, verticalment, o en diagonal. Una paraula *és dins del tauler* si es pot ubicar dins del tauler de manera que les lletres consecutives de la paraula apareguin a caselles adjacents del tauler. Atenció perquè, a diferència de les sopes de lletres clàssiques, no tota la paraula ha d'estar ubicada en la mateixa direcció (horitzontal, vertical o diagonal). Això és el que la fa estrambòlica.

L'objectiu del joc és trobar, en el menor temps possible, totes les paraules vàlides que s'amaguen a la Super Sopa. Una paraula és vàlida si s'hi troba al diccionari.

## 2 Objectius

L'objectiu d'aquest projecte és analitzar el cost i l'efectivitat de diferents algorismes i estructures de dades per a resoldre el problema de la Super Sopa.

Pel que fa a algorismes, s'haurà d'implementar la forma de cercar paraules dins del tauler i consultar si són paraules correctes, és a dir, si són al diccionari. Pel que fa a estructures de dades, s'haurà de decidir com implementar el diccionari per tal que aquestes consultes es facin de la manera més eficient possible.

S'han de realitzar les següents diferents implementacions del diccionari:

- (a) Amb un vector ordenat
- (b) Amb un trie —alguna de les seves variants, incloent Patricia o *ternary search trees* ([6, cap. 28], [5, §6.3],[1])

- (c) Amb un filtre de Bloom [2, 7]
- (d) Amb una taula de hash amb *double hashing* ([6, cap. 9], [5, §6.4, Algorithm D])

La nota màxima que es pot obtenir al projecte dependrà de quantes d'aquestes estructures s'hi treballin. Fer només les dues primeres (vector i trie) permet optar a una nota màxima de 5 punts. Fer una tercera implementació, per filtre de Bloom (opció (c)) o per taula de hash (opció (d)), permet optar a un màxim de 7.5 punts. Per poder obtenir un màxim de 10 punts, cal haver treballat les quatre implementacions proposades.

## Experimentació

S'ha de realitzar un estudi experimental significatiu per tal d'analitzar el cost i efectivitat de les opcions treballades. Us proposem procedir de la següent forma:

Donat un diccionari  $\mathcal{D}$  que us proporcionarem, seleccioneu aleatòriament un subconjunt  $\mathcal{P} \subseteq \mathcal{D}$  de 20 paraules (és a dir,  $|\mathcal{P}| = 20$ ). Cadascuna d'aquestes paraules, s'han de "plantar" al tauler de la Super Sopa a partir d'una posició aleatòria (però que permeti col·locar la paraula sencera). Recordeu que no és una sopa tradicional, sinó estrambòtica. La resta de posicions del tauler s'emplenaran escollint aleatòriament lletres de l'alfabet. Pel fet de plantar les paraules de  $\mathcal{P}$  sabem que, com a mínim, el nostre algorisme n'hauria de trobar aquestes vint. Recordeu, però, que l'objectiu del joc es trobar *totes* les paraules vàlides que s'amaguen dins de la Super Sopa.

El tauler de la Super Sopa sempre és quadrat,  $n \times n$ . Assumirem que  $n \geq 10$ . Els vostres experiments han de considerar diferents mides de tauler i veure quin és el comportament dels algorismes per resoldre el joc. El conjunt  $\mathcal{P}$  de paraules a plantar en cadascun d'ells ha de ser sempre el mateix.

Els experiments que considerin la implementació del diccionari amb filtres de Bloom hauran d'estudiar la precisió de les solucions proporcionades (donat que el filtre de Bloom pot donar falsos positius).

## 3 Detalls de l'entrega

El nivell de sofisticació i esforç dedicat a la pràctica és opcional i es tindrà en compte a l'hora d'avaluar-la. Tingueu en compte que la documentació entregada ens ha de permetre valorar el nivell d'assoliment de la competència transversal que hem d'avaluar: **Capacitat d'autoaprenentatge**. En el context del projecte hi ha molts aspectes rellevants relacionats amb aquesta competència: des de l'estudi de nous algorismes i estructures de dades, fins al disseny i anàlisi d'experiments, i la documentació d'aquests tipus de treballs de recerca. La qualificació final del projecte reflectirà la qualitat del vostre aprenentatge, de l'experimentació feta i de la documentació. La qualitat del codi entregat (programes) es pressuposa i representarà una part molt petita de la qualificació final.

La documentació ha de recollir i presentar la feina feta, les fonts que s'han consultat, el que heu après i els resultats de l'experimentació. En particular és molt important que reflecteixi de forma succinta el que heu après. Si no es compleix aquesta condició, la qualificació final del projecte reflectirà la qualitat de la presentació i no es tindrà en consideració la resta de material lliurat.

## Què cal lliurar

Caldrà lliurar (via el Racó de la FIB) una carpeta comprimida (.zip) que contingui tots els programes necessaris per a compilar i executar els experiments que proposeu, els benchmarks utilitzats pels experiments, i la documentació (en format pdf) del projecte. L'informe ha de documentar acuradament els algorismes desenvolupats, les estructures de dades, els experiments realitzats i les conclusions que se'n deriven. La qualitat d'aquest informe és el factor més important per a la qualificació final del projecte.

En particular, cadascuna de les implementacions del diccionari haurà d'estar en un fitxer diferent i clarament identificable, per exemple:

- Un fitxer `diccSortedVector.cc` per a la implementació (a).
- Un fitxer `diccTrie.cc` per l'apartat (b).
- Un fitxer `diccBloomFilter.cc` per l'apartat (c).
- Un fitxer `diccDHashing.cc` per l'apartat (d).

## 4 Alguns punters

Aquest document és intencionadament vague i s'espera que investigueu pel vostre compte tot el que necessiteu per tal de poder resoldre el problema de la Super Sopa; en particular, les diferents estructures de dades que permeten implementar el diccionari de paraules que, per manca de temps, no veiem aquest quadrimestre a GRAU-A. Hi ha bibliografia magnífica al respecte i no us costarà gens trobar-ne informació. A mode d'ajuda inicial (i apart dels llibres [3, 4] de la bibliografia bàsica de l'assignatura), us mencionem aquí alguns punters d'informació que us poden ser d'utilitat per començar la vostra recerca:

## Referències

- [1] Jon L. Bentley and Robert Sedgewick. Ternary search trees, 1998. On-line: <https://www.drdobbs.com/database/ternary-search-trees/184410528#>.
- [2] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970. On-line: <https://dl.acm.org/doi/10.1145/362686.362692>.
- [3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009.
- [4] Jon Kleinberg and Éva Tardos. *Algorithm Design, Pearson New International Edition*. Pearson Education Limited, 2013.
- [5] Donald E. Knuth. *The Art of Computer Programming: Searching and Sorting (vol. 3)*. Addison-Wesley, Reading (MA), USA, 2nd edition, 1997.
- [6] Dinesh P. Mehta and Sartaj Sahni, editors. *Handbook of Data Structures and Applications*. Chapman & Hall/CRC, Boca Raton (FL), USA, 2005.
- [7] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.